Novembre 1984 - Lire 3.000 - Anno 1° - N. 5 - Sped Abb. post. gr. III/70 - Distr. MePe

COMMODORE

MENSILE PER UTENTI DI VIC 20 - C64 - C16 - PLUS 4

LA PROGRAMMAZIONE STRUTTURATA 2a PUNTATA

◆

CONTABILITA' FAMILIARE

◆

CORNUCOPIA

◆

DIETA

PROGRAMMI VALIDI E TESTATI ANCHE PER IL C16 E IL PLUS-4

Per motivi indipendenti dalla nostra volontà, una percentuale superiore alla media delle cassette di Commodore Club n. 2 presenta difficoltà di caricamento.

...E A DICEMBRE

RADDOPPIA !!!

Per evitare disagi ai nostri videolettori, Commodore Club n. 2 verrà duplicato a dicembre all'interno del terzo numero.
Per tutti, *quindi, Commodore Club n. 3 sarà un numero doppio!!!*
Tra gli altri meravigliosi giochi in edicola a dicembre: Oroscopo, Pinocchio, Mr. Chomp, Shuttle, Visitors, Panettone... e tanti altri.

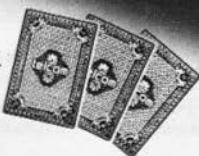
RESISTENTI, COMPATTE SILENZIOSE, EFFICIENTI, AFFIDABILI E COMPETITIVE.

Stampanti MITSUI con le caratteristiche tipiche dei giapponesi.

La TELCOM propone una gamma di stampanti che si distingue per la varietà delle funzioni e per la grande affidabilità:

- 80 e 132 colonne
- 120 e 180 caratteri al secondo bidirezionale ottimizzata
- fogli singoli, moduli continui
- vari tipi di caratteri
- stampa espansa, compressa, NLQ, grafica
- interfaccia parallela, seriale, buffer fino a 128 KB, interfaccie speciali.

adverteam

gioca la carta
telcom

Telcom s.r.l. 20148 Milano - Via M. Civitali, 75 - Tel. 4047648 (3 linee ric. aut.) - Telex 335654 TELCOM I

COMMODORE

DIRETTORE RESPONSABILE
Agostina Ronchetti

REDATTORE CAPO
Gloriano Rossi

REDAZIONE
Eugenio Coppari; Marco De Martino

SEGRETARIA DI REDAZIONE
Maura Ceccaroli

COLLABORATORI
Giancarlo De Cobelli; Marco De Rosa; Valerio Ferri; Francesco Gatti; Mariangela Guardione; Giulio Marcozzi; Mauro Massetti; Ernesto Sidoti; Renzo Zonin.

GRAFICA e IMPAGINAZIONE
Villa Iris s.n.c.
P.zza Massari, 8 - Milano

FOTO
Franco Vignati

DIFFUSIONE E ABBONAMENTI
Marina Vantini

DIREZIONE, REDAZIONE
Viale Famagosta, 75
20142 Milano - Tel. 02/8466675
Autorizzazione del Tribunale di Milano n. 103 del 25/2/84

STAMPA
Litografica - Busto Arsizio

Concessionario esclusivo per la diffusione - MEPE spa
Via G. Carcano, 32 - Milano

Spedizione in abbonamento postale - Gruppo III/70

Prezzo della rivista L. 3.000
Numero arretrato L. 6.000

Abbonamento annuo L. 25.000
I versamenti vanno indirizzati a: Commodore C.C.
V.le Famagosta, 75 - 20142 Milano, mediante emissione di assegno bancario, utilizzando il c/c postale n.ro 31532203

Per i cambi di indirizzo, indicare, oltre naturalmente il nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500 anche in francobolli.

LA POSTA

• Vorrei che pubblicaste un circuito che permette il collegamento in parallelo di una batteria con effetto tampone in caso di black-out. La rivista Nuova Elettronica N. 96 ha pubblicato qualcosa del genere però il circuito non funziona perché manca l'alimentazione di 9 Vac necessaria al C64.

In attesa di una vostra cortese risposta ringrazio anticipatamente. *(Giuseppe de Caterina)*

☐ Quale appassionato di elettronica ho tutte le raccolte di Nuova Elettronica e naturalmente anche il numero 96 è in mio possesso. Ho subito notato l'articolo in questione e ne ho apprezzata l'idea.

L'articolo di Nuova Elettronica N. 96 a pagina 28 va proprio bene in particolar modo con il VIC 20, mentre per il Commodore 64 occorre aspettare l'errata carrige che c'è sul numero 97 della rivista Nuova Elettronica, a pagina 116.

In queste righe è spiegato anche perché la tensione di 9 Vac in corrente alternata non sia necessaria per la salvaguardia dei programmi e dati.

• È possibile pubblicare il listato di un programma che velocizzi il caricamento da nastro? (per il Commodore 64). Se sì grazie. *(Gualtieri Giorgio)*

☐ Esiste sul mercato un prodotto, in due versioni (l'una per il Commodore 64 e l'altra per il VIC 20), che serve proprio per velocizzare la lettura e la scrittura da nastro.

Si chiama TURBO-TAPE. A tutt'oggi è considerato il migliore velocizzatore, quindi senza alcun dubbio ne consiglio l'acquisto.

• Ho fatto un programma per il "lotto" e vorrei personalizzarlo per non farmelo copiare. Mi potreste dire come devo fare? *(Antonio Mantino)*

☐ *Ho già avuto modo di ribadire che non esiste un modo veramente infallibile di proteggere programmi.*

Per certi è sufficente la compilazione, per altri invece sono necessari artifizi più o meno complicati che vanno da routines particolari alle chiavi hardware di diversa fattura.

Cosa fare allora? Consiglio di rivolgersi ad un softwarista bravo in LM, al quale affidare il suo programma. Così a distanza altro non si può suggerire.

• Poiché sul libretto di istruzioni è spiegato male come si registra un programma su un nuovo floppy disk, vorrei che mi spiegaste bene il funzionamento. Grazie e complimenti per la rivista. *(Carlo Ianelli)*

☐ Partiamo proprio dall'inizio:

1— Prendere un dischetto nuovo ed inserirlo in maniera corretta nel drive

2— digitare OPEN15, 8, 15, "NO: nome programma, NN": CLOSE15 seguito dal tasto RETURN. Il nome del disco è a piacimento e le due "NN" stanno per due lettere o numeri che corrispondono all'identificatore, il vero e proprio nome per il DOS. Dopo circa 80 secondi si potrà proseguire (cioè dopo che sarà riapparso il cursore sullo schermo ed il drive si sarà fermato).

A questo punto abbiamo preparato il dischetto in modo tale che possa ospitare uno o più programmi.

3— Salvataggio di un programma su disco preparato: digitare: SAVE "nome programma", 8 seguito dal tasto RETURN. Il drive si metterà in movimento, e dopo un poco terminerà il salvataggio.

4— Verificare che l'operazione sia ansata a buon fine con:

VERIFY "nome programma", 8 seguito dal tasto RETURN.

• Vorrei sapere quale stampante mi consigliate tra quelle della Seikosha: la GP 500 A, GP 700 A, e la GP 100 VC/ITALY? Io vorrei comprarmi un VIC 20 e così insieme compresi la stampante da voi consigliata. *(Emanuele Gianturco)*

☐ Dato che la decisione di acquisto è verso un VIC 20 non posso che far presente che esiste l'MPS 801, oppure la 1525 entrambe Commodore. Queste stampanti sono sì Commodore, ma la meccanica e parte dell'elettronica sono Seikosha. Quale scegliere allora?

Non posso rispondere altro che: un accessorio originale è sempre meglio di uno non originale.

• Vorrei soddisfare una curiosità: sul manuale del C64 che ho il connettore audio-video ha solo 5 poli, mentre sul mio C64 ne ho 3 in più Perché? *(Claudio Salviato)*

☐ Non servono proprio a nulla!! Ma allora, perché? "Mi chiederà ancora lei. Per avere una risposta sicura occorrerebbe andare in California e ripetere la domanda a chi ha compilato la "PART LIST".

Noi possiamo ipotizzare qualche utilizzo aggiuntivo, poi non messo in pratica, oppure è stato più conveniente acquistare questo connettore piuttosto che il normale 5 poli acquistare questo connettore piuttosto che il normale 5 poli a 180".

• Sono un possessore di Commodore 64, e vi sarei grato se mi spediste un programma per sproteggere i listati. *(Roberto Maldari)*

☐ Ogni giorno si cerca un metodo nuovo per proteggere i propri programmi e ...ogni giorno varie persone cercano di sproteggere gli altrui lavori.

A parte il fatto che non è bene sproteggere, così come ci sono vari metodi ed artifizi di protezione, altrettanti possono essere i sistemi di sprotezione.

• Possiedo un C64 ed un TV Blaupunkt che purtroppo dopo circa 15 minuti dall'accensione perde il colore.

Premetto che: il TV funziona regolarmente con l'ordinaria antenna per programmi televisivi, che il C64 produce gli stessi effetti dopo una prolungata accensione collegandolo al TV, che il TV dopo la stessa perde il colore quasi istantaneamente.

Vorrei un aiuto su come celebrare finalmente e felicemente la loro "unione". *(Mari Giuseppe)*

☐ I sintomi denunciati non lasciano dubbi.

Il televisore con buone probabilità è un modello un po' vecchiotto e questo fatto si puo' ripercuotere sul circuito elettronico chiamato CAS (controllo Automatico di Sintonia) che non riesce a mantenere il giusto canale sia quando arriva un segnale normale che, a maggior ragione, quando il segnale in questione è di notevole intensità.

Il C64 fornisce una notevole intensità di segnale, quindi una soluzione potrebbe essere quella di porre in serie, cioè fra C64 ed TV, un attenuatore. In ogni caso il sistema migliore è quello di avvalersi di un buon tecnico TV.

LA POSTA LA I

Una sola riga

Dato l'enorme successo che già dal primo annuncio ha avuto questa iniziativa si è deciso di dedicare all'argomento una Rubrica a se stante dal titolo omonimo.

Tutti i lettori che invieranno programmi costituiti da una sola riga, come ad esempio:

```
IL MOSCHINO PAZZO

    1 A$="[UP][LEFT][DOWN][RIGHT]":PRI
      NT"[LEFT][RVS] "MID$(A$,RND(1)*4
      +1,1)"[LEFT][RVOFF]*";:FOR I=1 T
      O 30:NEXT:GOTO 1
```

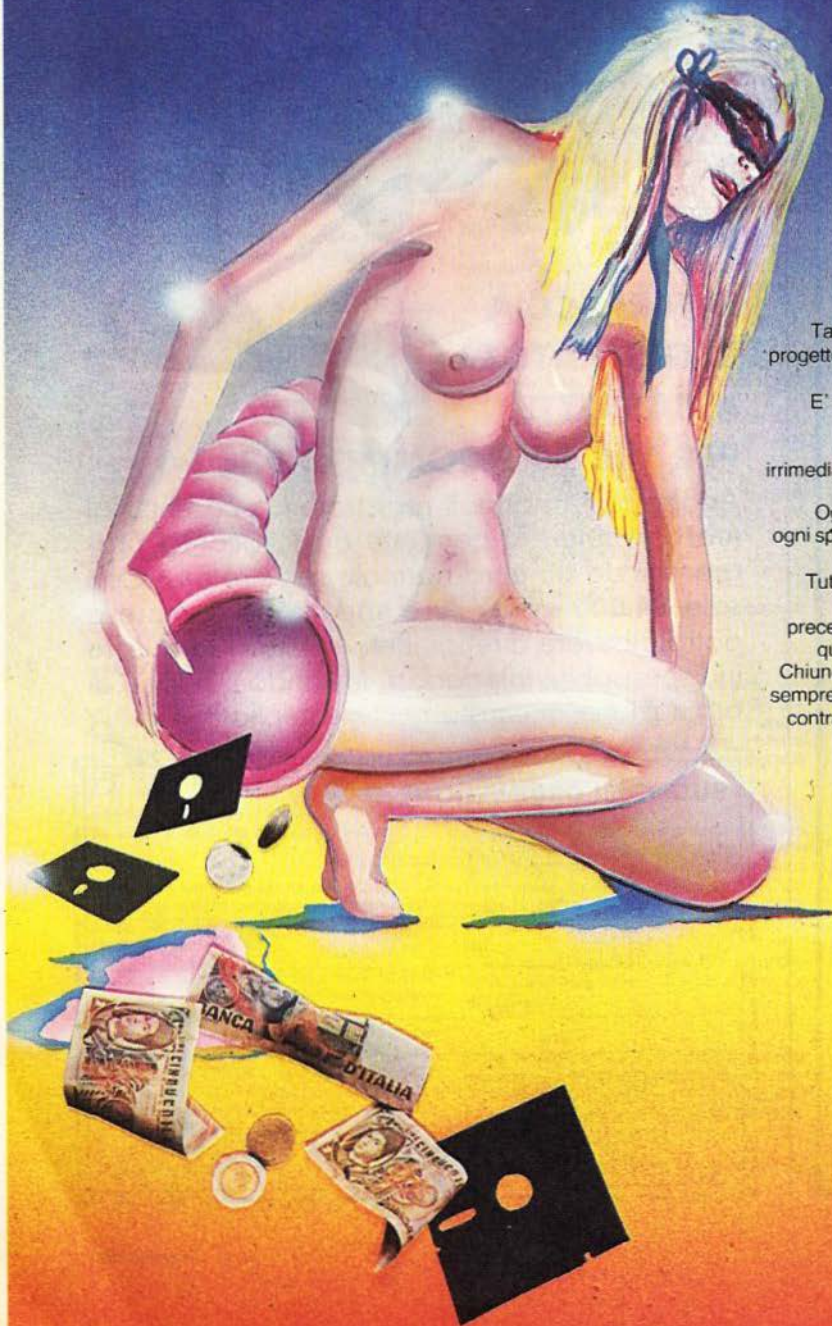
che verranno pubblicati sulla rivista saranno ricompensati con un libro a scelta tra:

Programo in Basic di **Clizio Merli**
64 programmi per il Commodore 64 di **Gloriano Rossi**
Utility e routine per il Commodore 64 di **Gloriano Rossi**
Come scegliere un Computer di **Michele di Pisa**
Progettare col Computer Edizioni **Minisystems**
Il Basic per tutti Edizioni **Minisystems**
Il linguaggio Pascal di **Clizio Merli**
Il linguaggio ADA Edizioni **Minisystems**
CP/M facile Edizioni **Minisystems**
100 software per progettare Edizioni **Minisystems**
1994: il mondo dei computer amici di **Alberto Cultrera**
JetSim Edizioni **Systems**
Il linguaggio Fortran di **Clizio Merli**

I programmi potranno avere uno scopo oppure no. Ciò non importa! I lavori dovranno pervenire solo su carta con una piccola descrizione, citando il nome, l'indirizzo ed il libro scelto. Tutti i programmi saranno presi in dovuta considerazione.

a cura di **Gloriano Rossi**

Cornucopia

Tante volte è la piccola idea che fa grande un progetto. Quante volte due parole ben scelte sono valse più di mille in un lungo discorso?
E' tranquillizzante, per il neofita, sapere che il 1541 quando da quel classico rumore: "tactactactac" non si sta rompendo irrimediabilmente, ma anzi sta eseguendo l'azione di sincronismo con la traccia zero.
Ogni idea, ogni curiosità, ogni suggerimento, ogni spigolatura software può essere valida e può trovare posto in Cornucopia.
Tutti i "doni" della dea Fortuna sono numerati con il sistema esadecimale. Sul numero precedente abbiamo iniziato da $01 fino a $11 e quindi oggi riprendiamo logicamente da $12.
Chiunque abbia una idea scriva ed ogni $nn sarà sempre firmato a meno che l'autore non desideri il contrario e... i migliori $nn saranno ricompensati adeguatamente.
Inviate i vostri $nn a:
Spett. Rivista COMMODORE
rubrica Cornucopia
Gloriano Rossi
Viale Famagosta 75
20142 Milano

In $05

Per un errore, nel primo numero di "Cornucopia" (nell'$05) è stato omesso il listato che qui riportiamo.

```
100 DIM TF(8)
110 TF(1)=133:TF(3)=134:TF(5)=135
    :TF(7)=136
120 TF(2)=137:TF(4)=138:TF(6)=139
    :TF(8)=140
200 GET X$:IF X$="" THEN 200
210 X=ASC(X$)
220 IF X<133 OR X>140 THEN 200
230 FOR I=1 TO 8:IF TF(I)=X THEN
    XI=I
240 NEXT
250 PRINT"E' STATO PREMUTO IL [RV
    S] TF   [3 LEFT]"XI
300 ON  XI GOTO  ...,...,...,...,
    ...,...,...,...
310 ON  XI GOSUB  ...,...,...,...
    ,...,...,...,...
```

$12

Una sinusoide variopinta. La locazione di memoria 646 ospita il valore numerico che va da 1 a 8 per il VIC 20, mentre per il C64 il contenuto varia da 1 a 16.

Questo numero è direttamente associato al colore definito attuale, cioè il colore che è stato attivato per ultimo. Questo semplice programma serve proprio a dimostrare tutto ciò:

Per il VIC 210:

10 POKE 646, RND (O)*8: PRINT TAB (SIN (X)*8+10) "[RVS]...": X=X+.3: GOTO 10

Per il C64:

10 POKE 646, RND (O)*16: PRINT TAB (SIN (X)*16+10) "[RVS]...": X=X+.3: GOTO 10.

$13

LOAD automatico da PRG - A volte può rendersi necessario, vuoi per comodità che per pigrizia, caricare un programma direttamente dal programma residente in memoria.

Uno dei sistemi più semplici e più comodi può considerarsi quello proposto qui. La numerazione delle righe, naturalmente, potrà essere cambiato in funzione delle esigenze del programma chiamante. Per chi avesse esclusivamente il registratore, sarà sufficiente sostituire nella riga 300, in fondo, il numero 8 con il numero 1.

```
300 PRINT"[4 DOWN]LOAD"CHR$(34)"N
    OME PRG"CHR$(34)",8[4 DOWN]"
305 PRINT"RUN[2 DOWN]"
306 PRINT"REM    ATTENDERE·PREGO"
310 POKE 631,13:POKE 632,13:POKE
    633,13:POKE 198,3
320 PRINT"[12 UP]"
```

$14

Da destra a sinistra. Questo piccolo programma ci permette di scrivere da destra verso sinistra, carattere per carattere, una determinata stringa in maniera tale che sembri scorrere nel medesimo verso.

```
100 A$ = "CORNUCOPIA"
110 FOR I = LEN (A$) TO 1 STEP-1
120: PRINT TAB (1)" "MID$(A$, I,1);"[UP]"
130: FOR X = 1 TO 10: NEXT X
140 NEXT I
150 GOTO 110
```

$15

Pallina a caso. Ecco un esempio di come si possa far muovere un carattere grafico sullo schermo senza ricorrere a sprites o a routines in linguaggio macchina.

```
 10 REM *******************
 20 REM *                 *
 30 REM * MOVIMENTO DI UNA *
 40 REM * PALLINA A CASO   *
 50 REM *                 *
 90 REM *******************
100 PRINT"[CLEAR][2 DOWN]"
110 X=INT(RND(1)*8):IF X=Y THEN 1
    10
120 A=INT(RND(1)*15):B=INT(RND(1)
    *15):PRINT"[LEFT]  ";
130 ON XGOSUB 140,160,150,170,160
    ,150,170,140=Y=X:GOTO 110
140 FOR K=1 TO A:PRINT"[LEFT] [DO
    WN]●";:GOSUB 190:NEXT:RETURN
150 FOR K=1 TO B:PRINT"[LEFT] [DO
    WN][2 LEFT]●";:GOSUB 190:NEXT
    :RETURN
160 FOR K=1 TO A:PRINT"[LEFT] [UP
    ][2 LEFT]●";:GOSUB 190:NEXT:R
    ETURN
170 FOR K=1 TO B:PRINT"[LEFT] [UP
    ]●";:GOSUB 190:NEXT:RETURN
180 GOTO 110
190 DL=10:FOR I=1 TO DL:NEXT:RETU
    RN
```

$16

Sposta su TV. Poter spostare il cursore in maniera diretta è cosa alquanto semplice con i tasti controllo, ma se questa azione è necessaria all'interno di un programma può sembrare un po' più complicata. Ecco quindi una routine atta proprio a questo scopo.

Per adattare la routine al VIC 20 occorre fornire la locazione di memoria di video meno 1 alla riga 250, e quindi dalla riga 260 alla riga 340 sostituire, dopo le PEEK, il numero 10 con il numero 17, il 23 con il 26, quindi il 62 con il 48, poi il 14 con il 49, il 12 con il 33, il 20 con il 34 ed infine il 13 con il 41.

E quindi alla 280, 290 il 40 con il 22; alla 300 e 330 il 41 con 23 ed alla 310 e 320 il 39 con 21. Ed infine alla riga 350 e 360 il numero 1000 con il numero 506.

```
100 REM ********************
110 REM * SPOSTA SUL VIDEO     *
120 REM * SECONDO I TASTI :    *
130 REM *                      *
140 REM *       Q W E          *
150 REM *       A S D          *
160 REM *       Z X C          *
170 REM *                      *
180 REM * VERSIONE PER C-64    *
190 REM *                      *
200 REM ********************
210
220 PRINT"[CLEAR]"
230 A=500
240
250 POKE 1023+A,81
260 IF PEEK(197)=10 THEN POKE 102
    3+A,32:A=A-01:REM  A SINISTRA
270 IF PEEK(197)=18 THEN POKE 102
    3+A,32:A=A+01:REM  A DESTRA
280 IF PEEK(197)=23 THEN POKE 102
    3+A,32:A=A+40:REM  GIU'
290 IF PEEK(197)=09 THEN POKE 102
    3+A,32:A=A-40:REM  SU'
300 IF PEEK(197)=62 THEN POKE 102
    3+A,32:A=A-41:REM  SU A SINIS
    TRA
310 IF PEEK(197)=14 THEN POKE 102
    3+A,32:A=A-39:REM  SU A DESTR
    A
320 IF PEEK(197)=12 THEN POKE 102
    3+A,32:A=A+39:REM  GIU A SINI
    STRA
330 IF PEEK(197)=20 THEN POKE 102
    3+A,32:A=A+41:REM  GIU A DEST
    RA
340 IF PEEK(197)=13 THEN POKE 102
    3+A,32:A=0500:REM  CENTRO
350 IF A>1000 THEN A=A-1000
360 IF A<1 THEN A=A+1000
370 GOTO 250
```

$17

Hard Copy. Può essere necessario, a volte, eseguire la copia su carta di ciò che appare sul video. Il sistema migliore, in BASIC, è quello suggerito nelle righe del programma che segue.

Potrete inserire queste poche righe in un qualsiasi vostro programma quale routine da richiamare al momento opportuno, naturalmente dopo la CLOSE 4 della riga 210 si dovrà aggiungere un RETURN se detta ruotine sarà richiamata con un GOSUB.

```
 10 REM ****************
 20 REM * PRINT SCREEN *
 30 REM ****************
100 N=0
110 XT$=""
120 OPEN 4,4
130 FOR I=1024 TO 2023
140 :X=PEEK(I):IF X=32 OR X=160 T
    HEN 160
150 :X=(X AND 127) OR ((X AND 64)
    *2) OR ((64-X AND 32)*2)
160 :X$=CHR$(X)
170 :XT$=XT$+X$
180 :N=N+1
190 :IF N>39 THEN PRINT#4,XT$:XT$
    ="":N=0
200 NEXT
210 CLOSE 4
```

$18

Lo sparacaratteri. Questo programma non ha una utilità pratica vera e propria. Infatti può essere paragonato ad un vero e proprio gadget, (un gadget è in genere un oggetto piacevole che non serve proprio a nulla).

Ciò nonostante può essere inserito in un altro programma per arricchirlo di un effetto aggiuntivo piacevole. Dopo aver introdotto una stringa di caratteri questa viene "sparata" letteralmente sul video accompagnata con il tipico colpo di una pistola.

```
100 GOTO 220
110 " ┌──────────────────────┐
120 " |                      |
130 " | LO SPARACARATTERI    |
140 " |                      |
150 " |          DI          |
160 " |                      |
170 " |  GLORIANO ROSSI      |
180 " ├──────────────────────┤
190 " |  DA UNA IDEA DI      |
200 " | RICCARDO  SAETTI     |
210 " └──────────────────────┘
220 SI=54272:FL=SI:FH=SI+1:TL=SI+
    2:TH=SI+3:W=SI+4:A=SI+5:H=SI+
    6:L=SI+24
230 W$="*":FOR X=1 TO 40:W$=W$+"*
    ":NEXT
240 DE$="":FOR I=1 TO 20:DE$=DE$+
    "[RIGHT]":NEXT
```

```
250 REM *****************
260 REM * INIZIO E INPUT *
270 REM *   STRINGA      *
280 REM *****************
290 PRINT"[CLEAR][8 DOWN]INSERISC
    I UNA STRINGA LUNGA AL MASSIM
    O"
300 PRINT"36 CARATTERI"
310 INPUT A$:LA=LEN(A$):IF LA>36
    THEN 290
320 REM *****************
330 REM * PRESENTAZIONE  *
340 REM * E RAFFICA      *
350 REM *****************
360 PRINT"[CLEAR][8 DOWN]"W$;SPC(
    38)"**"SPC(38)"**"SPC(38);W$"
    [4 UP]"
370 PRINTLEFT$(DE$,20-LA/2);
380 FOR X=1 TO LEN(A$):Q$=MID$(A$
    ,X,1):PRINTQ$;:IF Q$=" " THEN
     400
390 GOSUB 470:REM  JSR SONORO
400 NEXT:PRINT:PRINTSPC(254)"[3 D
    OWN]PREMI [RVS]SPAZIO[RVOFF]"
    ;
410 IF PEEK(197)=60 THEN 290
420 GOTO 410
430 REM *****************
440 REM * ROUTINE SONORA *
450 REM *   DELLO SPARO  *
460 REM *****************
470 FOR P=15 TO 0 STEP -1
480 POKE L,P:POKE A,15:POKE H,0:P
    OKE FH,40:POKE FL,200:POKE W,
    129
490 NEXT
500 POKE W,0:POKE A,0
510 RETURN
```

$19

Un sort: SHELL. Fra i vari tipi di ordinamento che si possono utilizzare, quello del metodo SHELL è fra i più veloci: Provare per credere e... utilizzatelo.

```
100 REM *********************
110 REM * ORDINAMENTO       *
120 REM * SECONDO IL METODO *
130 REM *   S H E L L       *
140 REM *********************
150 DIM A(500)
160 PRINT"[CLEAR]SORT TIPO SHELL[
    DOWN]"
170 INPUT "QUANTI NUMERI (2-500)"
    ;NN
180 IF NN<2 OR NN>500 THEN END
190 FOR I=1 TO NN
200 :A(I)=RND(1)*NN
210 NEXTI
220 REM *********************
230 TI$="000000":PRINT"INIZIO ";T
    I$
240 D=NN:SW=0
250 D=INT((D+1)/2)
260 FOR N=1 TO NN-D
270 :IF A(N)<=A(N+D) THEN 300
280 T=A(N):A(N)=A(N+D):A(N+D)=T
290 SW=1
300 NEXTN
310 IF SW=1 THEN SW=0:GOTO 260
320 IF D>1 THEN 250
330 A$=TI$:PRINT"FINE    "A$
340 FOR I=1 TO NN:PRINTA(I):NEXT
350 PRINT"FINE    "A$
```

$1 A

LOAD da disco. Caricare un programma da disco sembra la cosa più facile di questa terra. Capita però a volte di digitare male il nome di ciò che vogliamo oppure abbreviamo il nome con l'asterisco senza badare che c'era già un altro programma che iniziava con le medesime lettere.

Per evitare una cattiva digitazione del nome o per evitare di far troppe manovre, ciò per i pigri, vi consiglio di abituarvi ad agire in questa maniera: dopo aver listato la directory del disco, ci si posiziona esattamente a fianco del nome del programma che desideriamo. Si, proprio dove ci sono i numerini dell'occupazione in blocchi.

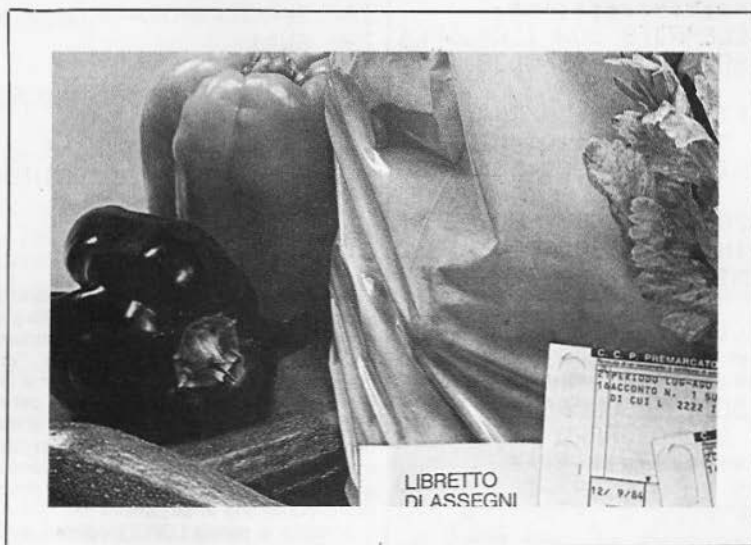
Si digita la parola LOAD (oppure L e O shiftato) e quindi con il controllo cursore ci si porta a destra del nome, proprio dopo i doppi apici e si digita ",8" seguito da tanti spazi quanti sono necessari per togliere la parola PRG.

La cancellazione della parola PRG non è necessaria se noi dovessimo caricare un programma con l'opzione ",8,1" oppure se abbiamo l'accortezza di aggiungere un due punti dopo l'otto (",8:" o ",8,1:").

$1 B

PEEK = RND. Per ottenere un numero casuale che può andare da 0 a 255 si esegue normalmente questo tipo di istruzione:
N = INT (RND(—1)*256)
Provate allora ad eseguire:
N = PEEK(162)
e guarda caso si ottiene ancora un numero casuale da 0 a 255.
Se si volessero solo numeri compresi fra 0 e 15 si esegue:
N = PEEK (162) AND 15
Mentre se si volessero solo numeri pari, sempre compresi fra 0 e 256, si fa:
N = PEEK (162) AND 254.

CONTABILITÀ FAMILIARE

di **Francesco Gatti**

L'uso del computer nella gestione e controllo della casa non dovrebbe essere una cosa tanto strana: dopotutto la stragrande maggioranza dei computer presenti nelle nostre case vengono definiti "Home Computer".

Elaboratori perciò che non solo sono rivolti a tutti (se così interpretiamo la parola 'Home' riferita al prezzo e alla sua semplicità d'uso), ma rivolti anche a risolvere problematiche esclusive della casa.

Da qui la concezione di un computer che dovrebbe aiutare i bambini nell'apprendimento dei problemi di matematica e di grammatica, aiutare la padrona di casa a tenere in ordine i suoi conti, controllare apparecchiature elettriche funzionanti in orario economico e così via.

Se analizziamo però i programmi per i vari VIC 20 o C64 ci accorgiamo che gran parte di questi sono rappresentati da "giochini". Vengono poi le "utilities" (come vengono chiamati i programmi che migliorano o semplificano talune proprietà del Computer), poi, quasi fanalini di coda, troviamo i cosiddetti gestionali.

In questo ultimo gruppo sono riuniti tutti i programmi che servono a velocizzare, semplificare operazioni di riordino archivi, di contabilità.

Programmi di questo tipo ad esempio per il C64, sono un po' pochi, fatta eccezione per le rubriche telefoniche (nella mia raccolta di programmi ne ho ben 5, tutti simili), dove secondo me la velocità del computer (compreso reperimento programma, caricamento, inserimento dati) viene ridicolizzata al confronto della velocità della mano e dell'occhio che consultano una rubrica convenzionale. Perciò quando si trattò di realizzare qualcosa che fosse utile avere nella memoria del nostro computer di casa, tenni conto di questo fattore (velocità). Scelsi così di dedicarmi al problema della contabilità giornaliera.

Il programma

All'esperto di contabilità parrà molto difficile che un programma di questo tipo possa trovare posto, per via della sua mole, su una rivista. E infatti non sbaglia: quello che viene trattato ora è un condensato di un più grosso program-

ma da me già realizzato per la rivista su cassetta "Commodore Club" della medesima casa Editrice e che verrà diffuso prossimamente.

Questa versione risulta semplificata riguardo alla ricerca particolareggiata delle voci (per campi, come: tipo di pagamento, codice del conto, ammontare totale di un conto, ammontare relativo a tutti i conti ecc. ecc.), la stampa generale e la sua gestione in maschere.

Nonostante ciò rimane pur sempre un valido strumento per documentare le nostre spese.

Come funziona

Il programma è stato studiato per funzionare su tutti i tipi di computer COMMODORE. L'unica raccomandazione riguarda i possessori del VIC 20 che dovranno DIMensionare le matrici della riga 186 secondo la loro disponibilità di memoria.

Il supporto di memoria di massa può essere indifferentemente il disco oppure il registratore. In quest'ultimo caso si devono modificare le linee **462** e **496** in:

462 OPEN 5, 1, 1, + (M$)
496 OPEN 5, 1, 0, + (M$)

In questa maniera non sarà necessario cambiare per le operazioni di CLOSE, PRINT#, INPUT#, l'IFN cioè il numero di riferimento interno. Lanciato il programma apparirà il menu principale. Seguendo le indicazioni poste in negativo (in REVERSE) si otterrà l'esecuzione delle varie operazioni.

Così ad esempio premendo la lettera "I" entreremo in "Modo Inserimento" permettendoci l'aggiornamento delle voci. Noterete che nell'introduzione delle voci esiste una riga di INPUT con la dicitura Codice Categoria: per risparmiare memoria occorrerà inserire TE per quanto riguarda le spese TElefoniche e così via.

Fare attenzione ad usate l'opzione C per la "Cancellazione", infatti l'indice delle voci slitterà indietro di una unità!

Le altre opzioni non creano ulteriori problemi, tranne forse solo Richiamo e Memorizzazione. Un consiglio: usate NOV84 per memorizzare il vostro FILE dati riguardanti il mese di novembre 1984.

Le tre POKE di riga 182 sono riferite al colore: le prime due riguardano lo schermo, la terza il colore del testo.

I possessori di computer delle serie 4000/8000 devono ignorarle; i possessori del VIC 20 con espansione dovranno modificare le prime due in POKE 36869,5 e POKE 36870,5.

```
PROGRAMMA : GEST.FAM PRG

-------*----------------- - - -
VAR.   ! LINEA DEL PROGRAMMA
-------*----------------- - - -
       !
A      !  520   550   560   570  1300  1350
       ! 1360  1370  1380  1420  1490
A$     ! 1460  1910
AG     !  720   750   760  1350  1380  1400
       ! 1660  1800  1810  1830  1840  1850
       ! 1860  1870  1890
AM$(   !  480   940  1150  1300  1360  1420
       ! 1490  1690  1850
B      !  590   600   610
B$(    !  240   250   260   270   280   290
       !  300   310   320   330   540   560
       ! 1550
C      !  690   700   710   820   830   840
       ! 1480
C$     !  490
CC$(   !  480   910  1120  1300  1360  1410
       ! 1480  1680  1840
D      !  850  1060  1310  1420  1510
DE$(   !  480  1000  1210  1310  1370  1420
       ! 1510  1710  1870
E      !  720   750   760   810   860   880
       !  910   940   970  1000  1070  1090
       ! 1120  1150  1180  1210
G      ! 1300  1410  1480
GM$    ! 1820  1830
GM$(   !  480   880  1090  1300  1360  1410
       ! 1480  1670  1830
H      ! 1540  1550  1560
I      ! 1260  1270  1300  1310  1350  1400
       ! 1410  1420  1440  1480  1490  1500
       ! 1510
I$(    !  340   350   360   370   380   390
       !  700   830
IN$    !  490   730  1020  1250  1320  1610
       ! 1760
K      !  630   640   650   660   670   680
       !  770   820  1240  1530
M      ! 1310  1420  1500
M$     ! 1620  1630  1650  1770  1780  1790
       ! 1920
MP$(   !  480   970  1180  1310  1370  1420
       ! 1500  1700  1860
N      ! 1580  1590
OP$    !  240   600
P      ! 1400  1430  1470  1900
R$     !  580   600   620   630   640   650
       !  660   670   740   750  1030  1040
       ! 1330  1340  2010  2020
ST     ! 1880
T      ! 1660  1670  1680  1690  1700  1710
       ! 1730
T$(    !  400   410   420   430   440   450
       !  500   680   770   820  1240  1530
       ! 1920
VO     !  790   800   810
```

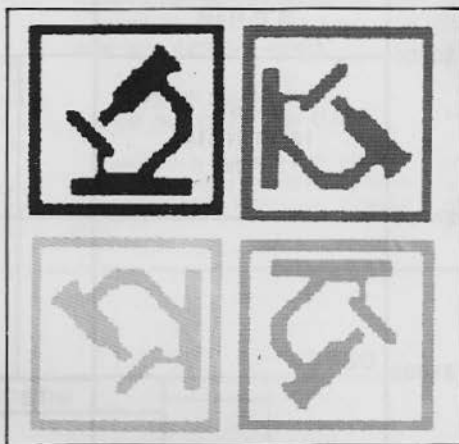
```
100 REM ****************************
102 REM * CONTABILITA' FAMILIARE    *
104 REM ****************************
106 REM * AUTHOR SOFTWARE:          *
108 REM * FRANCESCO GATTI           *
110 REM ****************************
112 REM * VIC 20              NO    *
114 REM * VIC 20+EXP          SI    *
116 REM * COMMODORE 64        SI    *
118 REM * C16                 SI    *
120 REM * PLUS 4              SI    *
122 REM * SERIE 4000/8000     SI    *
126 REM ****************************
128 REM * STRINGHE MESSAGGI         *
130 REM ****************************

132 B$(0)="[AZZUR][2 RIGHT]MENU PRI
    NCIPALE:CONTABILITA' ":OP$="ICN
    SRF"
134 B$(1)="[2 DOWN][GIALLO][RVS] I
    [RVOFF] - [RVS]I[RVOFF][BIANCO]
    NSERIMENTO VOCI"
136 B$(2)="[GIALLO][RVS] C [RVOFF]
    - [RVS]C[RVOFF][BIANCO]AMBIO VO
    CI "
138 B$(3)="[GIALLO][RVS] N [RVOFF]
    - [BIANCO]CA[GIALLO][RVS]N[RVOF
    F][BIANCO]CELLAZIONE VOCI"
140 B$(4)="[GIALLO][RVS] S [RVOFF]
    - [RVS]S[RVOFF][BIANCO]OMMARIO
    VOCI "
142 B$(5)="[GIALLO][RVS] R [RVOFF]
    - [RVS]R[RVOFF][BIANCO]ICHIAMO
    E MEMORIZAZIONE"
144 B$(6)="[AZZUR][RVS] F [RVOFF] -
     [RVS]F[RVOFF][BIANCO]INE"
146 B$(7)="[2 DOWN][GIALLO][RVS] 1
    [RVOFF][BIANCO] MEMORIZZAZIONE"
148 B$(8)="[GIALLO][RVS] 2 [RVOFF][
    BIANCO] RICHIAMO"
150 B$(9)="[GIALLO][RVS] 3 [RVOFF][
    BIANCO] MENU PRINCIPALE"
152 I$(0)="[3 DOWN]       VOCE NUMER
    O : "
154 I$(1)="  GIORNO DEL MESE : [RVS
    ]  [2 LEFT][RVOFF]"
156 I$(2)=" CODICE CATEGORIA : [RVS
    ]  [2 LEFT][RVOFF]"
158 I$(3)="          AMMONTARE : [RVS
    ]          [9 LEFT][RVOFF]"
160 I$(4)="MODO DI PAGAMENTO : [RVS
    ]     [4 LEFT][RVOFF]"
162 I$(5)="        DESCRIZIONE : [RVS
    ]              [12 LEFT][RVOFF]"
164 T$(0)="CONTABILITA' FAMILIARE"
166 T$(1)="[DOWN][RVS][GIALLO] MODO
     INSERIMENTO [RVOFF][BIANCO]"
168 T$(2)="[DOWN][RVS][GIALLO] MODO
     CAMBIO VOCI [RVOFF][BIANCO]"
170 T$(3)="[DOWN][RVS][GIALLO] MODO
     CANCELLAZIONE [RVOFF][BIANCO]"
172 T$(4)="[DOWN][RVS][GIALLO] SOMM
    ARIO VOCI PER [RVOFF][BIANCO] "
174 T$(5)="[DOWN][RVS][GIALLO] RICH
    IAMO E MEMORIZZAZIONE [RVOFF][B
    IANCO]"

176 REM ****************************
178 REM * INIZIO PROGRAMMA          *
180 REM ****************************
182 PRINT"[CLEAR]":POKE 53280,6:POK
    E 53281,6:POKE 646,1
184 OPEN 1,0
186 DIM GM$(100),CC$(50),AM$(50),MP
    $(50),DE$(50)
188 C$=CHR$(13):IN$=CHR$(157)
190 PRINT:PRINT:PRINT TAB(6)T$(0)
192 PRINT TAB(8)"[2 DOWN]DI GATTI F
    RANCESCO"
194 FOR A=0 TO 500:NEXTA

196 REM ****************************
198 REM * MENU PRINCIPALE           *
200 REM ****************************
202 PRINT"[CLEAR]"
204 PRINT TAB(3)B$(0)
206 FOR A=1 TO 6
208 PRINT TAB(6)B$(A):PRINT
210 NEXTA
212 GET R$:IF R$="" THEN 212
214 FOR B=1 TO 7
216 IF MID$(OP$,B,1)=R$ THEN 220
218 NEXTB:GOTO 212
220 IF R$="F" THEN 532
222 IF R$="I" THEN K=1:GOTO 238
224 IF R$="C" THEN K=2:GOTO 262
226 IF R$="N" THEN K=3:GOTO 362
228 IF R$="S" THEN K=4:GOTO 398
230 IF R$="R" THEN K=5:GOTO 432
232 REM ****************************
234 REM * INSERIMENTO VOCI          *
236 REM ****************************
238 PRINT"[CLEAR]":PRINT TAB(8)T$(K
    )
```

```
240 FOR C=0 TO 6
242 PRINTI$(C):PRINT
244 NEXTC
246 E=AG:E=E+1:GOSUB 320
248 PRINT"[2 DOWN][RVS][GIALLO] VUO
    I CONTINUARE ? [RVOFF][BIANCO]:
     SI"IN$IN$;
250 INPUT#1,R$
252 IF R$="NO" THEN AG=E:GOTO 202
254 AG=E:GOTO 246

256 REM ***************************
258 REM * MODIFICA VOCI           *
260 REM ***************************
262 PRINT"[CLEAR]":PRINT TAB(8)T$(K
    )
264 PRINT"[2 DOWN][RVS][GIALLO] QUA
    LE VOCE DESISERI CAMBIARE ? [RV
    OFF][BIANCO] ";
266 INPUT#1,VO
268 IF VO=0 THEN 202
270 E=VO
272 PRINT"[CLEAR]":PRINT TAB(8)T$(K
    ):FOR C=0 TO 6
274 PRINTI$(C):PRINT
276 NEXTC
278 PRINT"[HOME]":FOR D=0 TO 4:PRIN
    T:NEXTD
280 PRINT TAB(20)E:PRINT
282 PRINT TAB(20)
284 PRINTGM$(E)
286 PRINT
288 PRINT TAB(20)
290 PRINTCC$(E)
292 PRINT
294 PRINT TAB(20)
296 PRINTAM$(E)
298 PRINT
300 PRINT TAB(20)
302 PRINTMP$(E)
304 PRINT
306 PRINT TAB(20)
308 PRINTDE$(E)
310 PRINT:GOSUB 320
312 PRINT"[2 DOWN][RVS][GIALLO] VUO
    I CONTINUARE ? [RVOFF][BIANCO]:
     SI"IN$IN$;
314 INPUT#1,R$
316 IF R$="SI" THEN 262
318 GOTO 202
320 PRINT"[HOME]":FOR D=0 TO 4:PRIN
    T:NEXTD
322 PRINT TAB(20)E:PRINT
324 PRINT TAB(20);
326 INPUT#1,GM$(E)
328 PRINT:PRINT
330 PRINT TAB(20);
332 INPUT#1,CC$(E)
334 PRINT:PRINT
336 PRINT TAB(20);
338 INPUT#1,AM$(E)
340 PRINT:PRINT
342 PRINT TAB(20);
344 INPUT#1,MP$(E)
346 PRINT:PRINT
348 PRINT TAB(20);
350 INPUT#1,DE$(E)
352 PRINT
354 RETURN

356 REM ***************************
358 REM * CANCELLAZIONE VOCI      *
360 REM ***************************
362 PRINT"[CLEAR]":PRINT TAB(8)T$(K
    )
364 PRINT"[2 DOWN][RVS][GIALLO] QUA
    LE VOCE VUOI CANCELLARE ? [RVOF
    F][BIANCO]: 0"IN$;
366 INPUT#1,I
368 IF I=0 THEN 202
370 GOSUB 416
372 PRINT:PRINT"[2 DOWN][RVS] GG.[R
    VOFF] [RVS] COD.[RVOFF] [RVS] A
    MMONT.[RVOFF] [RVS] MODP.[RVOFF
    ] [RVS] DESCRIZIONE [RVOFF]"
374 PRINT TAB(4-G)GM$(I); TAB(7)CC$
    (I); TAB(20-A)AM$(I);
376 PRINT TAB(25-M)MP$(I); TAB(40-D
    )DE$(I)
378 PRINT:PRINT"[2 DOWN]CONFERMI LA
     CANCELLAZIONE ? :NO"IN$IN$;
380 INPUT#1,R$
382 IF R$="NO" THEN 202
384 FOR A=I TO AG
386 GM$(A)=GM$(A+1):CC$(A)=CC$(A+1)
    :AM$(A)=AM$(A+1)
388 MP$(A)=MP$(A+1):DE$(A)=DE$(A+1)
390 NEXTA:AG=AG-1:GOTO 202

392 REM ***************************
394 REM * SOMMARIO VOCI           *
396 REM ***************************
398 GOSUB 522
400 FOR I=1 TO AG:P=P+1:GOSUB 416
```

```
402 PRINTI; TAB(6-G)GM$(I); TAB(8)C
    C$(I);
404 PRINT TAB(21-A)AM$(I); TAB(26-M
    )MP$(I); TAB(40-D)DE$(I)
406 IF P>7 THEN GOSUB 518
408 NEXTI
410 PRINT"[DOWN] PREMI UN TASTO PER
     CONTINUARE "
412 GET A$:IF A$="" THEN 412
414 P=0:GOTO 202
416 G=LEN(GM$(I)):C=LEN(CC$(I))
418 A=LEN(AM$(I))
420 M=LEN(MP$(I))
422 D=LEN(DE$(I))
424 RETURN
426 REM ***************************
428 REM * GESTIONE ARCHIVIO DATI   *
430 REM ***************************
432 PRINT"[CLEAR]":PRINT TAB(4)T$(K
    )
434 FOR H=7 TO 10
436 PRINT TAB(6)B$(H):PRINT
438 NEXTH
440 PRINT"[DOWN] COSA SCEGLI ? ";
442 INPUT#1,N
444 ON NGOTO 452,488,202

446 REM ***************************
448 REM * MEMORIZZAZIONE DATI      *
450 REM ***************************
452 PRINT"[CLEAR]":PRINT TAB(6)"[RV
    S][GIALLO] MEMORIZZAZIONE [RVOF
    F][BIANCO]"
454 PRINT"[2 DOWN][RVS] PER CHE MES
    E E ANNO ? [RVOFF]:MMMAA"IN$IN$
    IN$IN$IN$;
456 INPUT#1,M$
458 IF M$="MMMAA" THEN 202
460 PRINT:PRINT"[2 DOWN][RVS][GIALL
    O] SCRITTURA DATI [RVOFF][BIANC
    O]"
462 OPEN 5,8,5,"@0:"+M$+",S,W"
464 FOR T=1 TO AG+1
466 PRINT#5,GM$(T)
468 PRINT#5,CC$(T)
470 PRINT#5,AM$(T)
472 PRINT#5,MP$(T)
474 PRINT#5,DE$(T)
476 NEXT
478 FOR T=1 TO 5:PRINT#5,"**":NEXT
480 CLOSE 5:GOTO 202
482 REM ***************************
484 REM * RICHIAMO DATI            *
486 REM ***************************
488 PRINT"[CLEAR]":PRINT TAB(13)"[R
    VS][GIALLO] RICHIAMO [RVOFF][BI
    ANCO]"
490 PRINT:PRINT"[2 DOWN][RVS] PER C
    HE MESE E ANNO ? [RVOFF]:MMMAA"
    IN$IN$IN$IN$IN$;
492 INPUT#1,M$
494 IF M$="MMMAA" THEN 202
496 OPEN 5,8,5,"0:"+M$+",S,R"
498 AG=0
500 AG=AG+1
502 INPUT#5,GM$:IF GM$="**" THEN 51
    6
504 GM$(AG)=GM$
506 INPUT#5,CC$(AG)
508 INPUT#5,AM$(AG)
510 INPUT#5,MP$(AG)
512 INPUT#5,DE$(AG)
514 IF ST=0 THEN 500
516 AG=AG-1:CLOSE 5:GOTO 202
518 P=0:PRINT"  PREMI UN TASTO PER
    CONTINUARE "
520 GET A$:IF A$="" THEN 520
522 PRINT"[CLEAR]":PRINT TAB(8)T$(4
    )M$
524 PRINT"[2 DOWN][RVS] N.[RVOFF] [
    RVS]GG.[RVOFF] [RVS]COD.[RVOFF]
     [RVS] AMMONT.[RVOFF] [RVS]MODP
    .[RVOFF] [RVS]DESCRIZIONE [RVOF
    F]":RETURN
526 REM ***************************
528 REM * USCITA PROGRAMMA         *
530 REM ***************************
532 PRINT"[CLEAR]":PRINT TAB(11)"[R
    VS] ATTENZIONE [RVOFF]"
534 PRINT:PRINT TAB(5)"[2 DOWN]PRIM
    A DI USCIRE DAL PROGRAMMA"
536 PRINT TAB(5)"[DOWN]ASSICURATI D
    'AVER SALVATO"
538 PRINT TAB(5)"[DOWN]I TUOI DATI
    "
540 PRINT TAB(5)"[DOWN]PREMI [RVS]
    M [RVOFF] PER TORNARE AL"
542 PRINT TAB(5)"[DOWN]MENU PRINCIP
    ALE, UN ALTRO TASTO"
544 PRINT TAB(5)"[DOWN]PER TERMINAR
    E"
546 GET R$:IF R$="" THEN 546
548 IF R$="M" THEN 202
550 PRINT"[CLEAR]":CLOSE 1:END
```

MICROSCOPIO

di Marco de Rosa

parte 1ª

In questo articolo, e negli altri che seguiranno, descriverò, locazione per locazione, quella zona della RAM usata dal sistema operativo del CBM 64. Questa parte dal byte 0 e giunge fino ai puntatori dei dati degli sprites, cioè al byte 2047 compreso. Certe zone saranno descritte in modo particolarmente dettagliato nel corpo dell'articolo, e saranno indicate in neretto nella mappa globale.

Prima di cominciare è opportuno dare un occhiata ai termini che verranno usati più frequentemente:

• **ASCII:** American Standard Code for Information Interchange. È un modo standard di assegnare dei numeri ai caratteri della tastiera. I computer Commodore hanno un codice molto simile all'ASCII anche se non proprio standard.

• **Default:** indica lo stato delle locazioni al momento dell'accensione della macchina

• **Pointer:** (puntatore). Con questo nome si indicano due bytes consecutivi che contengono l'indirizzo di un registro contenente un dato. Il valore dell'indirizzo varia da 0 a 65535. Il primo dei due (low byte) contiene la parte bassa. L'altro (high byte), la parte alta.

L'indirizzo si calcola quindi:

low byte + 256 * high byte

Il pointer dell'inizio della zona BASIC, per fare un esempio, è alle locazioni 43 e 44. Per trovare dove comincia questa zona basta scrivere

```
PRINT PEEK (43) + 256 * PEEK (44)
```

Il risultato in condizioni di default deve essere 2049.

• **Flag:** (segnalatore). È un registro che può assumere vari valori, corrispondenti all'avverarsi o meno di certe condizioni.

Come esempio possiamo considerare il flag all'indirizzo 10. Questo assume il valore 0 o 1 a seconda che sia in corso una operazione di Load o di Verify.

• **Vector:** (vettore): è molto simile ad un puntatore, solo che l'indirizzo rappresenta l'inizio di una routine in linguaggio macchina.

• **Kernal:** È il sistema operativo del CBM 64. (Insieme di puntatori a routines).

• **Registro:** È un'altro modo di chiamare una locazione di memoria. Può essere più lungo di un Byte.

• **Pagina:** Sono 256 Bytes di memoria. Il CBM 64 ha 256 pagine di RAM.

• **Buffer:** È una zona della RAM usata per immagazzinare dati.

• **I/O:** Input/Output. Sono tutti quei dispositivi con cui il CBM 64 dialoga con l'esterno. Nell'articolo si riferisce al VIC II, alle due CIA, al SID e più in generale ai dispositivi presenti nei 4 Kb che vanno da $D000 a $DFFF.

• **CIA:** Complex Interface Adapter. Sono i microprocessori dedicati alla comunicazione del computer con il mondo esterno. Nel CBM 64 sono due.

• **VIC II:** Video Interface Controller. È il microprocessore dedicato alla gestione dello schermo.

• **SID:** Sound Interface Device. È lo speciale microprocessore dedicato alla generazione dei suoni e dei rumori.

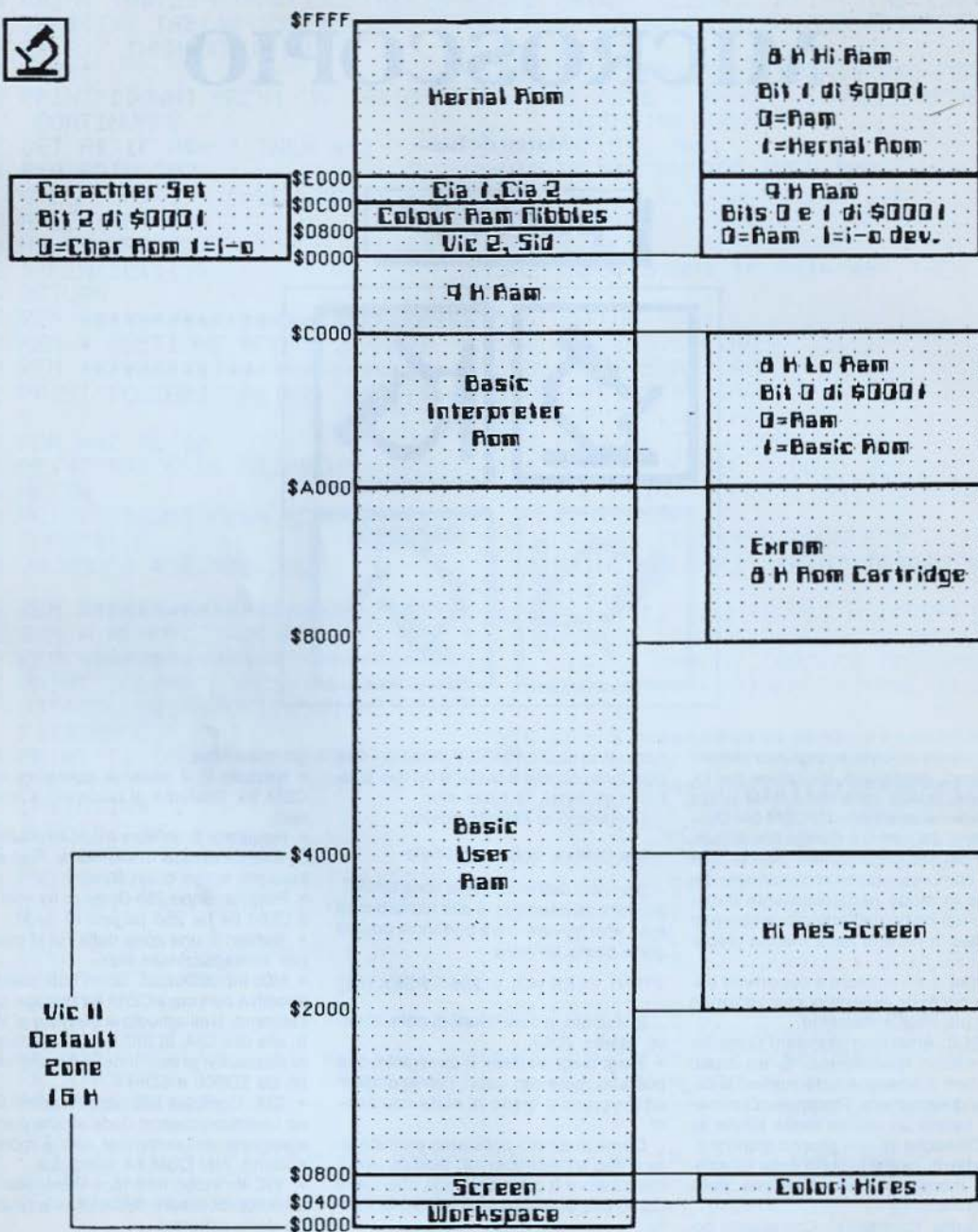



Figura 1: architettura della memoria del CBM 64

Bytes 0 e 1: area di gestione della memoria

Il CBM 64 riesce a gestire 64 Kb di RAM, 20 Kb di ROM e 4 Kb di dispositivi di I/O, usando un microprocessore, il 6510, capace di indirizzare solo 64 Kb. Il segreto sta nella porta di I/O locata all'indirizzo 1. Usando come interruttori i bit di questo registro è possibile accedere alle ROM "nascoste" sotto alla RAM principale. Il registro all'indirizzo 0 permette inoltre di definire la direzione del flusso di dati attraverso la porta di I/O.

Vediamo le caratteristiche dei bit della porta:

Bit 0: è chiamato LORAM. Gestisce gli 8 Kb della ROM del BASIC, da 40960 a 49151 ($A000-$BFFF). Se LORAM è 1 (High), la ROM è inserita. Se è 0 (Low) la ROM sparisce ed è rimpiazzata da 8 Kb di RAM.

Bit 1: è chiamato HIRAM. Gestisce gli 8 Kb della ROM del KERNAL, da 57344 a 65535 ($E000-$FFFF). Se HIRAM è High la ROM è inserita. Altrimenti al suo posto compaiono 8 Kb di RAM.

Bit 2: è chiamato CHAREN. Gestisce i 4 Kb della ROM dei caratteri, da 53248 a 57343 ($D000-$DFFF). Se CHAREN è High la ROM non è inserita e al suo posto compaiono la mappa dei colori di schermo e i dispositivi di I/O, che sono il VIC II, il SID e le due CIA.

Bit 3, 4 e 5: gestiscono il registratore a cassette. È importante non alterarne il contenuto.

Bit 6 e 7: non usati.

Il valore del registro direzione dati deve essere 47, cioè 00111101 binario, dove 1 indica un Output e 0 un Input. In figura 1 potete vedere l'architettura della memoria del CBM 64 con il riassunto delle cose dette fino ad ora.

Bytes da 43 a 56: area puntatori del BASIC

Questa zona contiene sette puntatori che gestiscono le zone di memoria occupate dal BASIC. I loro valori di default sono:

Bytes	Descrizione	default
43-44	Inizio area programmi	2049
45-46	Inizio area variabili	2051
47-48	Inizio area vettori	2051
49-50	Fine area vettori	2051
51-52	Inizio area stringhe	40960
53-54	Utilità generale	40960
55-56	Tetto del basic	40960

Nella figura 2 potete vedere come sono disposte queste zone in presenza del programma seguente.

```
10 REM GESTIONE AREA PROGRAMMI BASIC
20 A$ = "PIPPO": B$ = "TOPOLINO": C$=A$+B$
30 DIM A(20)
100 FOR I = 43 TO 55 STEP 2
110 PRINT PEEK(I)+256*PEEK(I+1)
120 NEXT I
```

Il programma stesso stampa sullo schermo i valori dei sette puntatori. Provate a modificarlo aumentando la lunghezza delle stringhe, delle merci, ecc. e confrontate i risultati ottenuti.

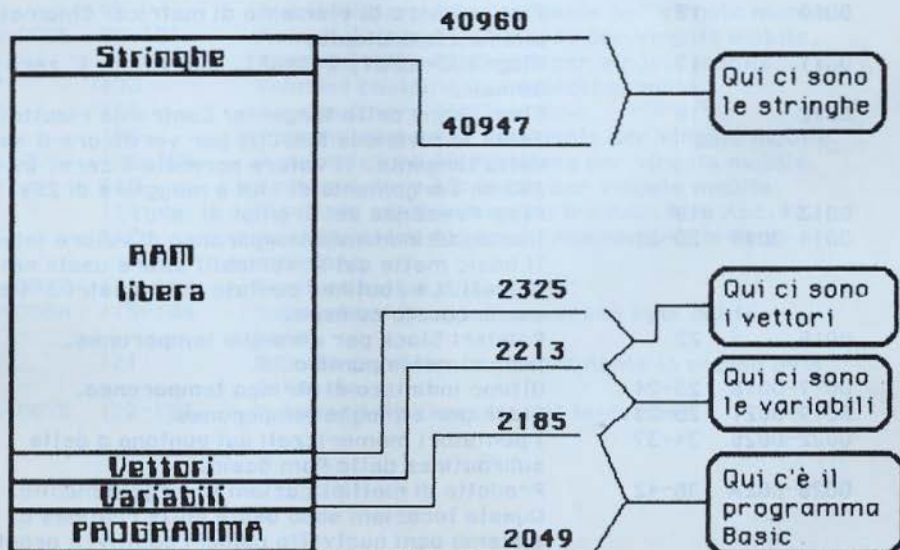



Figura 2: zone di memoria disposte in funzione del programma di pag.

HEX	DEC	DESCRIZIONE
0000	0	**Registro direzione dati del 6510**
0001	1	**Registro di I/O a 8 bit.**
0002	2	E' usato dal computer per memorizzare temporaneamente informazioni di cui ha bisogno. Non usatelo, potrebbe "inchiodare" la macchina.
0003-0004	3-4	Jump vector per la conversione da virgola mobile ad intero.
0005-0006	5-6	Jump vector per la conversione da intero a virgola mobile.
0007	7	Ricerca carattere. Memorizza il valore ASCII 34 (virgolette).
0008	8	Flag di ricerca virgolette alla fine di una stringa.
0009	9	Colonna schermo dall'ultimo TAB.
000A	10	Flag: 0=Load, 1=Verify. Il valore cambia con l'ultima operazione di disco o nastro.
000B	11	Pointer al buffer di input/ Numero di elementi.
000C	12	Flag: dimensionamento matrice di default. Contiene il valore della prima lettera dell'ultima matrice dimensionata.
000D	13	Tipo di dati in lettura: $FF=stringa, $00=numerico.
000E	14	Tipo di dati in lettura: $80=Intero, $00=virgola mobile.
000F	15	Flag: scansione DATA/ virgolette LIST/ Raccolta spazzatura. Il valore di default è 4.
0010	16	Flag: numero di elemento di matrice/ Chiamata di una funzione utente.
0011	17	Flag: $00=INPUT, $40=GET, $98=READ. E' zero di default.
0012	18	Flag: Segno della tangente/ Confronto risultati. Testa la divisione SIN/COS per verificare il segno della tangente. Il valore normale è zero. Diventa 255 se l'argomento di TAN è maggiore di 259.1.
0013	19	Flag: Presenza del prompt di INPUT.
0014-0015	20-21	Immagazzinamento temporaneo di valore intero. Il basic mette qui le variabili intere usate nei calcoli. Le routines puntate dai registri 3-4 e 5-6 usano questo numero.
0016	22	Pointer: Stack per stringhe temporanee. Normalmente punta a 25.
0017-0018	23-24	Ultimo indirizzo di stringa temporanea.
0019-0021	25-33	Stack per stringhe temporanee.
0022-0025	34-37	I puntatori memorizzati qui puntano a delle subroutines della Rom Basic.
0026-002A	38-42	Prodotto di moltiplicazioni in virgola mobile. Queste locazioni sono usate dalle routines di sistema ogni qualvolta numeri positivi o negativi molto grandi sono moltiplicati.
002B-002C	43-44	**Pointer: Inizio dell'area programmi basic. Default $801(2049)**

HEX	DEC	DESCRIZIONE
002D-002E	45-46	**Pointer: Inizio dell'area variabili basic.**
002F-0030	47-48	**Pointer: Inizio dell'area vettori basic.**
0031-0032	49-50	**Pointer: Fine dell'area vettori basic (+1).**
0033-0034	51-52	**Pointer: Inizio dell'area stringhe.**
0035-0036	53-54	**Pointer: Stringhe di utilità.**
0037-0038	55-56	**Pointer: Tetto del basic.**
0039-003A	57-58	Numero della linea basic corrente.
003B-003C	59-60	Numero della linea basic precedente.
003D-003E	61-62	Pointer: Indirizzo del Basic da cui ripartire dopo il CONT. Normalmente è 2048.
003F-0040	63-64	Numero della linea di DATA corrente.
0041-0042	65-66	Pointer: Indirizzo del dato corrente in una riga DATA. Normalmente è 2048.
0043-0044	67-68	Vettore: routine di Input. Questo è l'indirizzo della locazione dove l'INPUT mette temporaneamente i suoi dati, cioè i 78 bytes da 512 a 600. Non potete cambiare questo valore. Mettere piu di 78 bytes nel buffer porta alla perdita del buffer stesso!
0045-0046	69-70	Nome della variabile basic corrente. Quando il nome è un carattere singolo, contiene il suo valore ASCII.
0047-0048	71-72	Pointer: Variabile attualmente in elaborazione.
0049-004A	73-74	Pointer: Variabile di indice cicli FOR..NEXT.
004B-0060	75-96	Puntatori temporanei/ Area dati.
0061	97	Accumulatore #1: Esponente per virgola mobile.
0062-0065	98-101	Accumulatore #1: Mantissa per virgola mobile.
0066	102	Accumulatore #1: Segno per virgola mobile.
0067	103	Pointer: costante per calcoli in serie.
0068	104	Accumulatore #1: Overflow.
0069	105	Accumulatore #2: Esponente per virgola mobile.
006A-006D	106-109	Accumulatore #2: Mantissa per virgola mobile.
006E	110	Accumulatore #2: Segno per virgola mobile.
006F	111	Risultato confronto segni tra Acc.#1 e Acc.#2.
0070	112	Accumulatore #1: arrotondamento di ordine inferiore.
0071-0072	113-114	Pointer: buffer di cassetta.
0073-008A	115-138	Subroutine: Prende il prossimo byte del testo basic.
0079	121	Entrata per ottenere nuovamente lo stesso byte del testo.
007A-007B	122-123	Pointer: Byte corrente del testo basic.

PROGRAMMAZIONE STRUTTURATA

di **Mariangela Guardione**

(seconda parte)

In questo secondo articolo sulla programmazione strutturata, applicata al BASIC, si esamineranno parte delle istruzioni che costituiscono questo linguaggio (mentre le rimanenti verranno esaminate nelle successive puntate) da punto di vista di struttura logica.

È infatti necessario acquisire la capacità di programmare ovvero di analizzare e di formulare un problema in maniera rigorosa per ottenere l'abilità di affrontare in maniera metodologica tutte le tematiche legate alla programmazione, prescindendo dal tipo di linguaggio utilizzato.

BASIC è l'acronimo di "Beginner All-purpose Symbolic Instraction Code" che tradotto liberamente in itliano significa: "Linguaggio simbolico di programmazione adatto per i principianti per tutte le applicazioni". Questo linguaggio fu sviluppato negli anni 60 al Dartmouth College (Ohio, USA), come utilizzo interattivo per applicazioni scientifiche e per uso didattico.

Da principio esso fu principalmente utilizzato da ricercatori e da tecnici, ma con l'avvento prima dei personal computers, e in seguito degli home ha avuto uno sviluppo estremamente rapido (anche per le notevoli estensioni) che gli consente attualmente di gestire problematiche tipiche sia nell'ambito gestionale che in quello tecnico-scientifico.

Un programma in BASIC è costituito da una o più linee che vengono digitate dall'utente tramite la tastiera. Alla fine di ognuna di esse viene premuto il taso RETURN, che serve per segnalare all'interprete la fine di una linea di programma. Ciascuna di queste linee è preceduta da "un numero di identificazione". Rappresentando tutto questo in pseudo-codifica, di cui abbiamo parlato nella prima parte, si ha ciò che vediamo nella prima figura.

Il numero di identificazione serve affinché il computer riconosca l'ordine in cui devono essere eseguite le varie istruzioni. Volendo, l'utente potrebbe inserire le linee in un ordine qualsiasi. Infatti è l'interprete stesso che provvede automaticamente ad inserire in maniera ordinata tutte le istruzioni secondo una successione crescente, sia in visualizzazione che in esecuzione possiamo osservare la progressione delle linee.

Ecco perché questo numero di identificazione permette un'eventuale inserzione di una o più linee, all'interno del programma, utilizzando semplicemente un numero intermedio. Ad esempio:

```
5 PRINT "CALCOLO DEI QUADRATI e CUBI"
10 INPUT A
20 B = A*A
30 C = A*A*A
```

Ora possiamo inserire tre linee di print semplicemente digitando:

```
15 PRINT A
25 PRINT B
40 PRINT C
```

Otterremo, come certo sapete, questo risultato:

```
5 PRINT "CALCOLO DEI QUADRATI e CUBI"
10 INPUT A
15 PRINT A
20 B = A*A
25 PRINT B
30 C = A*A*A
40 PRINT C
```

Questo è il motivo per cui, normalmente, nella stesura di un programma in BASIC, viene impiegato un PASSO di numerazione in decine. Un altro utilizzo di questo identificatore è quello di riferimento, quando viene utilizzata un'istruzione

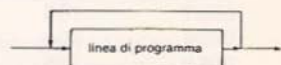
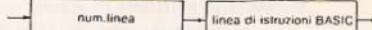
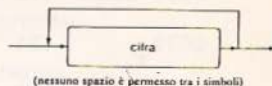
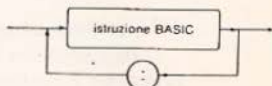
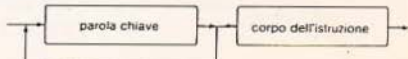



Figura 1: rappresentazioni in pseudo-codifica del numero di riga BASIC.

di salto incondizionato (istruzione che verrà di seguito esaminata).

Ciascuna linea di istruzione in BASIC può essere costituita da una o più istruzioni separate dal carattere ":".

A questo punto, prima di iniziare ad esaminare le varie istruzioni, è meglio spiegare cosa esse significhino in BASIC.

Un'istruzione BASIC è costituita da:

- *parola chiave:* è una parola riservata che specifica l'azione che si vuol compiere. (È buona norma non utilizzare mai variabili il cui nome sia simile ad una parola chiave poiché questo creerebbe confusioni in fase di elaborazione del programma).
- *Corpo dell'istruzione:* è rappresentato dall'insieme delle notazioni che descrivono in dettaglio gli oggetti su cui verrà compiuta l'azione decritta nel programma.

Istruzioni del BASIC

L'istruzione REM. REM rappresenta l'abbreviazione della parola REMARK che in inglese significa annotazione, commento.

Tutte le parole o i simboli che seguono REM vengono prese in considerazione dall'interprete BASIC non come appartenenti ad istruzioni eseguibili, ma bensì come autodocumentazione per le varie istruzioni che si vogliono far eseguire.

L'autodocumentazione dei programmi, come già spiegato nella prima parte, è molto importante in quanto permette al programmatore di ricordarsi tutto ciò che ha fatto eseguire nei programmi da lui realizzati anche a distanza di tempo.

L'istruzione REM nella notazione in psudo-codifica è rappresentata nella figura n. 2.

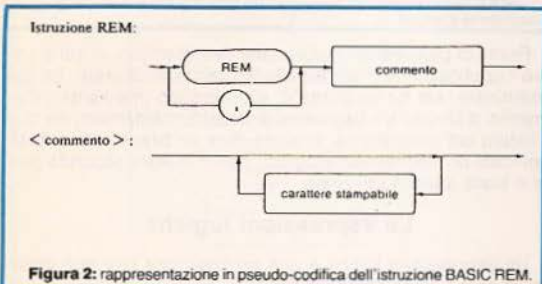



Figura 2: rappresentazione in pseudo-codifica dell'istruzione BASIC REM.

L'istruzione INPUT. Questa istruzione, il cui significato deriva dall'inglese IN = dentro, PUT = mettere, serve per leggere i dati dall'esterno (che nel caso dei personal l'INPUT classico è rappresentato dalla tastiera) e porli nella memoria dell'elaboratore alle posizioni che vengono identificate dai nomi delle variabili specificate nella lista che segue la parola chiave.

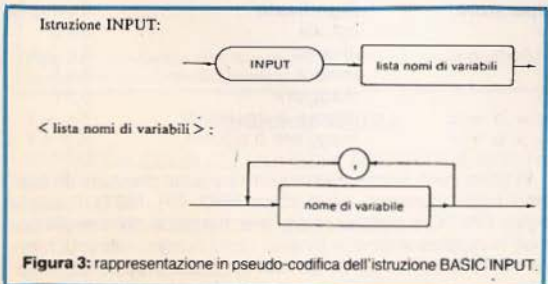



Figura 3: rappresentazione in pseudo-codifica dell'istruzione BASIC INPUT.

I valori delle precedenti variabili della lista vengono sostituiti dai nuovi e tutti i caratteri che vengono digitali attraverso la tastiera sono riemessi dall'elaboratore sullo schermo, affinché l'utente possa verificarne l'esattezza ed intervenire in caso di errore.

Anche per questa istruzione la notazione grafica in pseudo-codifica è rappresentata nella figura 3.

L'istruzione PRINT. L'istruzione PRINT, che in inglese significa stampa, serve per visualizzare i valori relativi alle variabili che compaiono nella lista che segue la parola chiave Nella lista di variabili sono possibili due separatori:

- la virgola che determina l'incolonnamento di ogni elemento della lista in una posizione predefinita
- il punto e virgola allinea i valori delle espressioni stampandoli l'uno di seguito all'altro sulla stessa riga.

Osserviamo nella figura n. 4 la pseudo-codifica per la PRINT.

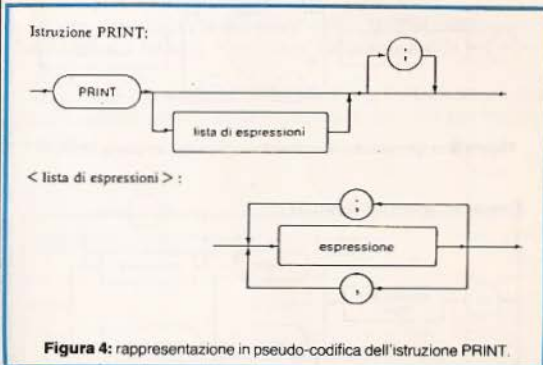



Figura 4: rappresentazione in pseudo-codifica dell'istruzione PRINT.

Fino a questo punto abbiamo esaminato le istruzioni relative all'immissione ed emissione dei dati in un programma. L'esecuzione di questo ipotetico programma può essere richiesta tramite il comando RUN seguito da un RETURN. A questo punto viene emesso il carattere "?", che invita l'utilizzatore ad inserire i dati, se questi dovranno essere più di uno allora saranno separati da una virgola e queste saranno tante quante sono le variabili che seguono la parola chiave INPUT.

Se

Sino ad ora sono state esaminate istruzioni eseguibili le une dopo le altre; ora si inizierà a trattare istuzioni che vengono eseguite, oppure no, in funzione del risultato VERO o FALSO di una proposizione logica e che servono per controllare il flusso logico del programma ; si vedrà infine, anche attraverso esempi, come si possano combinare queste istruzioni per ottenere programmi strutturati.

L'istruzione strutturata IF... THEN... Questa istruzione rappresenta una struttura alternativa in quanto permette al calcolatore di operare delle scelte. Dopo la parola IF si deve porre una condizione espressa mediante relazioni matematiche o logiche e quindi scrivere la seconda parola chiave che accompagna sempre IF, cioè il THEN, e quindi scrivere altre istruzioni.

Quando il calcolatore, durante l'esecuzione del programma, incontra l'istruzione IF, analizza la condizione che com-

pare dopo la parola chiave e decide se essa sia vera oppure falsa. A questo punto si comporta in maniera diversa in base alla conclusione cui esso è giunto. Se la relazione è vera allora esegue tutte le istruzioni che trova scritte dopo la parola THEN, altrimenti, conclude che la relazione esaminata è falsa e ignora totalmente tutte le istruzioni presenti sulla stessa linea di programma, che seguono il THEN, e continua l'esecuzione con la linea successiva a meno che l'interprete in funzione non preveda anche l'istruzione ELSE.

Con questa istruzione, come già detto precedentemente, è possibile far decidere al calcolatore se eseguire o meno alcune istruzioni basandosi su ciò che è stato eseguito precedentemente dal programma.

La rappresentazione grafica, mediante il diagramma sintattico, di questa istruzione è riportata nella figura n. 5.

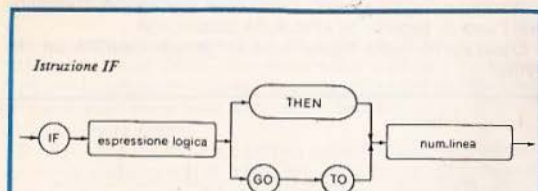



Figura 5: rappresentazione in pseudo-codifica dell'istruzione BASIC IF.

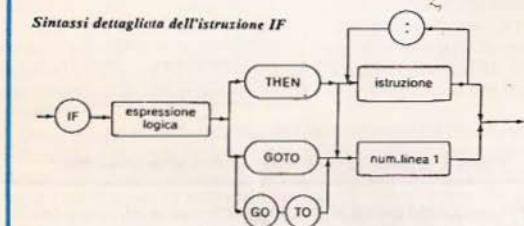



Figura 5a: rappresentazione in pseudo-codifica più dettagliata dell'istruzione BASIC IF.

L'istruzione strutturata GOTO. Questa istruzione è nota anche come "istruzione di salto incondizionato"; questo perché serve per mandare il calcolatore ad una qualsiasi linea di programma contenente l'istruzione che si vuole far eseguire. Adoperato all'interno di un programma, il GOTO permette al calcolatore di eseguire istruzioni che sono poste su linee di programma diverse in un ordine indipendnte dal numero di linea.

Anche per l'istruzione GOTO il diagramma sintattico è dato dalla rappresentazione di figura 6.

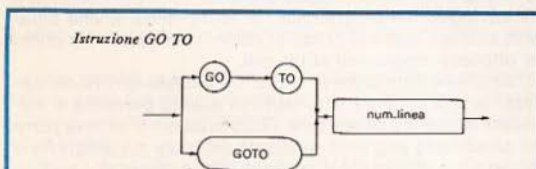



Figura 6: rappresentazione in pseudo-codifica di GOTO.

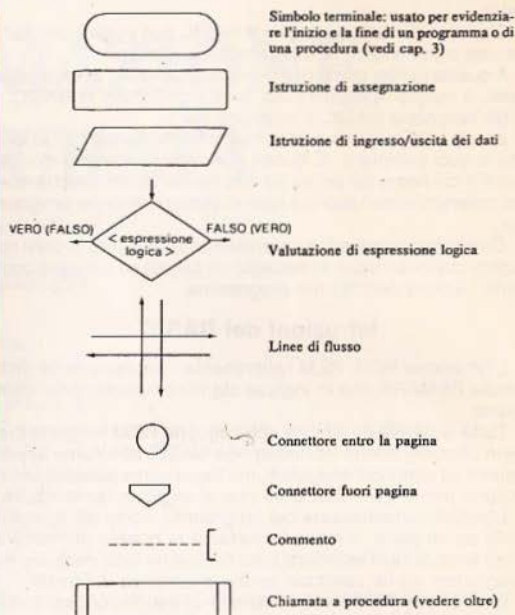



Figura 7: rappresentazioni e spiegazioni dei simboli grafici che compongono un diagramma a blocchi.

Prima di passare all'esposizione dell'esempio, in cui si vedrà l'applicaziome pratica delle istruzioni strutturate, fin qui esaminate, sia da un punto di vista grafico (mediante i diagrammi a blocchi e i diagrammi di Nassi/Shneiderman) sia con il listato del programma, si vuole dare un breve cenno al significato di "espressione logica" che in questa seconda parte è stata spesso utilizzata.

Le espressioni logiche

Un'espressione logica è una proposizione che può assumere solo i valori VERO e FALSO.

Esempio:
3<5: vero
20+20=50: falso

Negli esempi precedentemente elencati compaiono delle operazioni relazionali costituite da due operandi (costanti, variabili, valori di funzioni) legati da uno dei seguenti operatori:

Operatore	Significato	Esempio
=	uguale	3=3
< > o > <	diverso	3< >5
<	minore	3<5
>	maggiore	5>3
< = o = <	minore o uguale	X< =Y
= > o > =	maggiore o uguale	X> =Y

Vi sono, però, anche espressioni che sono costituite da operandi logici (operandi logici come AND, OR, NOT). Il valore logico FALSO è rappresentato internamente nel calcolatore (cioè in rappresentazione binaria) con il numero intero 0, mentre VERO con il valore 1. Questo permette di simulare varia-

bili logiche facendo assumere alle stesse soltanto i valori 0 e 1.

Si passa ora ad esaminare il programma esempio che è rappresentato da una serie di istruzioni che calcolano e stampano la media aritmetica di N numeri.

Definizione del problema:
Dato il numero N intero e gli N numeri reali della successione A1...An, acquisiti uno alla volta dalla tastiera, calcolare e stampare la media aritmetica degli stessi.

Analisi del problema:
- nome programma: MEDIA
- dati in input: N [variabile intera],
  A [variabile reale, acquisita una alla volta da tasiera]
- dati in output: M [media, variabile reale]
- elaborazioni: SOMMA = A1 + A2 + A3 + ...An

m = SOMMA/N

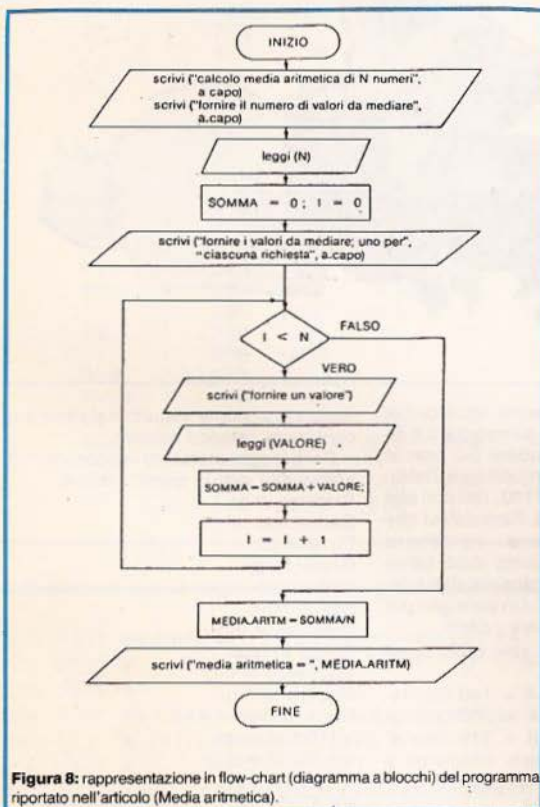



Figura 8: rappresentazione in flow-chart (diagramma a blocchi) del programma riportato nell'articolo (Media aritmetica).

Programma MEDIA

```
10 REM CALCOLO DELLA MEDIA ARITMETICA DI N NUMERI
20 INPUT N
30 REM INIZIALIZZAZIONE DELLA SOMMA
40 SOMMA = 0
50 REM INIZIALIZZAZIONE INDICE CONTROLLO
60 I=0
70 IF I=N THEN 130
80 REM INPUT DEI DATI
90 INPUT "VALORE DATO"; A
100 REM AGGIORNAMENTO SOMMA
110 SOMMA = SOMMA+A
120 I=I+1: GOTO 70
130 REM CALCOLO DELLA MEDIA
140 M = SOMMA/N
150 PRINT "LA MEDIA È :"; M
160 END
```

Si vuole ora rappresentare questo programma in forma grafica utilizzando, anziché la notazione lineare (che abbiamo trattato nella prima parte comparsa sulla rivista Commodore n. 4), altri due tipi di rappresentazione degli algoritmi che sono:

1 - Diagrammi a blocchi (Flow-Chart), i cui simboli sono visibili nella figura 7

La rappresentazione del programma MEDIA utilizzando il Diagramma a Blocchi è riportata nella figura 8

2 - Diagrammi di Nassi/Shneiderman, che forniscono una rappresentazione grafica delle istruzioni strutturate nel seguente modo:

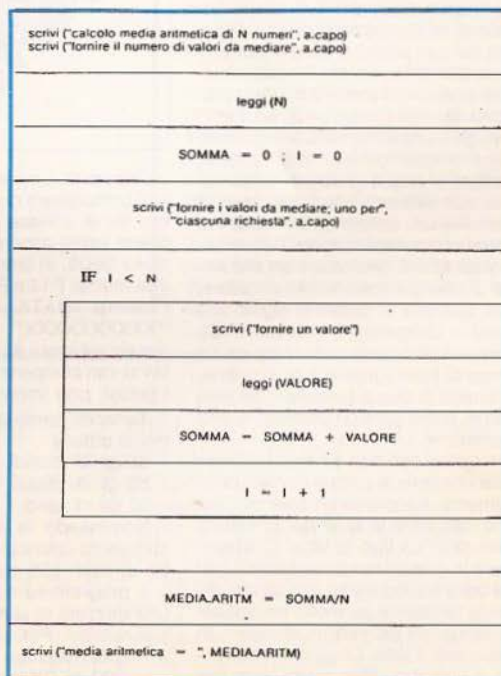



Figura 9: rappresentazione del medesimo programma MEDIA tramite il diagramma di Nassi/Shneiderman.

CONTROLLO NUTRITIVO DEI CIBI

di **Francesco Gatti**

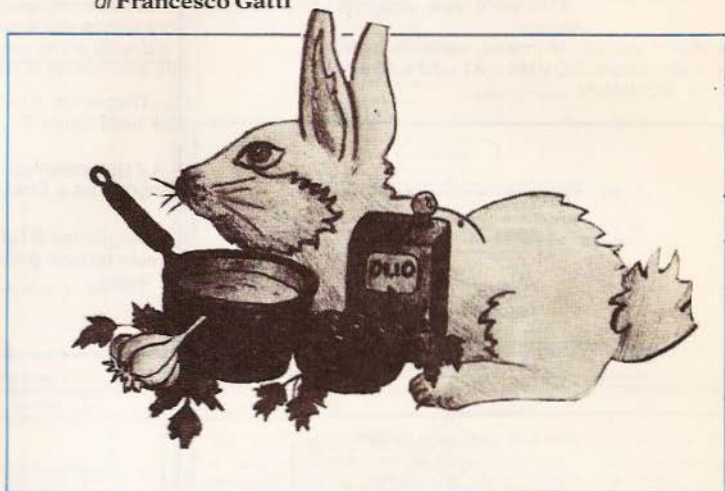
Sono sempre di più gli esperti che asseriscono che mangiamo molto e male. Riusciamo infatti a condire una sana spaghettata con elementi che, mischiati fra di loro, sono l'equivalente di un whisky a stomaco vuoto: un vero shock per il nostro apparato digerente.

Quante volte condiamo la nostra pastasciutta con il burro e poi aggiungiamo del sugo di pomodoro scaldato con dell'olio!. Penste al nostro fegato quando gli arriveranno insieme grassi animali e grassi vegetali...

Prima di continuare è doveroso fare una precisazione: il sottoscritto non si rende responsabile di danni alle persone causati da un uso non corretto o esagerato dei dati presenti nel programma trattato, quindi un consiglio spassionato: sottoponete ciò che vi propone il programma al vostro medico di fiducia.

Tutti gli alimenti di natura organica, contenenti cioè glicidi, lipidi, protidi, contengono acqua (H_2O) e carbonio. Questi due elementi sono combustibili per eccellenza, danno cioè reazioni di combustione e ossidazione. Come tutte le ossidazioni, provocano un alto sviluppo di energia sotto forma di calore, questa quantità di calore è legato alla quantità di idrogeno (H) e carbonio (C). La quantità di calorie sviluppate da un grammo di lipidi (grassi) è di 9 calorie, un grammo di glicidi (zuccheri) siluppa 4 calorie, come pure un grammo di protidi (proteine).

Con questi dati non dovrebbe essere difficile ottenere le calorie contenute in un alimento. Facciamo un esempio: vogliamo calcolare la quantità di calorie contenute in un litro di latte. Dovremo operare in questo modo: dobbiamo per prima cosa analizzare su una tabella alimentare (presente su molte enciclopedie) i composti del nostro alimento, in questo caso, il latte. Leggiamo che contiene, su un litro, 35 grammi di proteine, 50 grammi di lattosio (zucchero) e 35 grammi di grassi.

La terza opzione è chiara: visualizzerà tutti i cibi, con il peso e le calorie, da voi introdotti. La precisione dei valori dei vari componenti di un alimento è a 6 cifre. Se si volesse rendere più precisi questi valori occorre modificare l'istruzione MID$, in linea 1140, inerenti alle due matrici P1$ e P2$. Ricordatevi che l'ultima DATA deve contenere "XXXXXXXXXX", questo dato serve per riconoscere il termine dei dati relativi ai vari componenti. Un consiglio per i golosi: non imbrogliare i dati!!

Tenendo conto dei valori sopra riportati si ottiene:

35 gr. di protidi × 4 = 140 calorie
50 gr. di glicidi × 4 = 200 calorie
35 gr. di lipidi × 9 = 315 calorie

Sommando le calorie otteniamo il contenuto calorico complessivo di un litro di latte: 655 calorie.

Il programma contiene già i dati di una dozzina di alimenti con i loro principi nutritivi. Per ampliare il numero di alimenti, massimo 100 dato che alla linea 300 di DIMensionamento matrici abbiamo fornito un numero equivalente a 100, occorre introdurli nelle righe di DATA in coda al programma, in questo ordine però: il primo elemento corrisponde al nome seguito dal peso e il contenuto calorico relativo.

Poi per ogni elemento vengono forniti anche altri dati in questo ordine:

Proteine in gr.
Carboidrati in gr.
Fibre in gr.
Grassi in gr.
Polinsaturi in gr.
Sali in mgr.
Potassio in mgr.
Calcio in mgr.
Magnesio in mgr.
Fosforo in mgr.
Ferro in mgr.
Vit. B1 in mcgr.
Vit. B2 in mcgr.
Vit. pp in mgr.
Vit. C in mgr.
Vit. A in U.I.
Vit. D in U.I.

Per calcolare questi valori occorre eseguire dei calcoli in questa maniera:

Proteine = 15% delle calorie, perciò la sua formula è data da (cal*. 15)/4

Carboidrati = 55% delle calorie: (cal*. 55)/4

Fibre = praticamente carboidrati non

```
PROGRAMMA : PRINC-NUTR PRG

------*--------------- - - -
VAR.  ! LINEA DEL PROGRAMMA
------*--------------- - - -
      !
A     !  310   410   430   450   460   530
      !  550  1270  1280  1310  1350
A$    !  490   580   800  1110  1200  1210
      ! 1330
AL$(  !  300   330   460   470   930  1290
      ! 1310
B     !  340   450   470   530   550  1280
      ! 1300
C     ! 1180  1230  1240
CA    !  620   630   640   650   770
CA(   !  650
CC    !  320   330   340   350  1270  1310
G     ! 1230
I     !  420   530   710   860   870   880
      !  890   920  1020  1060
IN$   !  250   450
J     !  920   930   940  1020  1030  1040
      ! 1130  1140  1150  1240  1360
K     ! 1010  1030  1050
M     ! 1220  1230  1240
OPZ   !  530   710  1060  1130
P1$(  !  300   420   530   710  1130  1140
      ! 1150
P2$(  !  300   420   530  1060  1140  1150
PA(   !  300   310   340   460   470   530
      !  930  1030  1060  1230  1240
R$    !  740   750   820   830   950   960
      !  990  1000
RV$   !  550
RVS$  !  270
S1    ! 1280  1290  1310
SC(   !  300   420   870   880   930  1030
V     !  300
V(    !  630   640   650   660   670   680
      !  690   700   710
```

digeribili, varia da elemento a elemento

Grassi = 30% delle calorie: (cal*. 30)/9

Grassi polinsaturi = 35% dei grassi totali

Per quanto riguarda gli altri principi nutritivi non si può dare una formula di derivazione in quanto variano da elemento a elemento. Rimando alla consultazione di tabelle alimentari, oppure ad un amico medico.

In ogni caso troverete delle discordanze di valori con i calcoli delle formule, questo perché alcuni li ho ottenuti da tabelle già fatte.

Il programma

Le funzioni di questo programma sono trattate nel suo unico menu iniziale: confronto tra due alimenti, esame di più elementi, lista elementi inseriti.

La prima opzione si occupa di prendere due alimenti e confrontare i loro principi nutritivi permettendo così una scelta. Prima di confrontare questi due cibi contenuti nella lista totale, viene domandato all'utilizzatore se le porzioni sono di suo piacimento, in caso contrario ricalcola i principi nutritivi secondo la nuova quantità.

La seconda via di utilizzazione del programma è quasi come la precedente però su più alimenti. In questo caso esiste un piccolo vantaggio: introdotto il numero delle calorie necessarie in un giorno, il computer calcola tutti gli altri elementi contenuti in questo valore, che li comparerà con i valori dei cibi da voi inseriti. Sarà perciò facile determinare per ogni pasto la qualità e la quantità dei cibi che saranno più indicati per il nostro organismo.

```
100 REM ****************************
102 REM *   CONTROLLO NUTRITIVO    *
104 REM *         DEI CIBI         *
106 REM ****************************
108 REM * AUTHOR SOFTWARE:         *
110 REM *   FRANCESCO GATTI        *
112 REM ****************************
114 REM * VIC 20           NO      *
116 REM * VIC 20+EXP       SI      *
118 REM * COMMODORE 64     SI      *
120 REM * C16              SI      *
122 REM * PLUS 64          SI      *
124 REM * SERIE 4000/8000  SI      *
126 REM ****************************
128 IN$=CHR$(157)+CHR$(157)
130 POKE 53280,0:POKE 53281,0:POKE
    646,5:PRINT"[CLEAR]"
132 RVS$="[RVS]"
134 PRINT TAB(5)"[2 DOWN][RVS][GIAL
    LO] CONTROLLO NUTRITIVO DEI CIB
    I [RVOFF][VERDE]"
136 PRINT TAB(11)"[2 DOWN]DI FRANCE
    SCO GATTI"
138 REM ***************************
140 REM *      LETTURA DATI       *
142 REM ***************************
144 DIM AL$(100),PA(100,19),P1$(19)
    ,P2$(19),V (19),SC(20)
```

```
146 FOR A=1 TO 19:PA(0,A)=0:NEXTA
148 CC=1
150 READ AL$(CC):IF AL$(CC)="XXXXXX
    XXXX" THEN 162
152 FOR B=1 TO 19:READ PA(CC,B):NEX
    TB
154 CC=CC+1:GOTO 150
156 REM ***************************
158 REM * MENU PRINCIPALE          *
160 REM ***************************
162 PRINT"[CLEAR]"
164 PRINT TAB(5)"[2 DOWN][RVS][GIAL
    LO] 1 [RVOFF][VERDE] CONFRONTAR
    E DUE ALIMENTI"
166 PRINT TAB(5)"[DOWN][RVS][GIALLO
    ] 2 [RVOFF][VERDE] ESAMINARE DE
    GLI ALIMENTI"
168 PRINT TAB(5)"[DOWN][RVS][GIALLO
    ] 3 [RVOFF][VERDE] LISTA ALIMEN
    TI"
170 PRINT TAB(5)"[DOWN][RVS][GIALLO
    ] 4 [RVOFF][VERDE] FINE PROGRAM
    MA"
172 PRINT TAB(5)"[DOWN]COSA SCEGLI
    ";:INPUT A
174 FOR I=1 TO 19:P1$(I)="":P2$(I)=
    "":SC(I+1)=0:NEXTI
176 ON AGOTO 186,230,412,410
178 REM ***************************
180 REM *   VISUALIZZAZIONE        *
182 REM *     ALIMENTO A,B         *
184 REM ***************************
186 PRINT"[2 DOWN]QUALI ALIMENTI "
```

```
188 PRINT"[DOWN]DESIDERI CONFRONTAR
    E  : A,B "IN$IN$IN$;:INPUT A,B
190 PRINT"[CLEAR]":PRINTA; TAB(4);A
    L$(A); TAB(30);PA(A,1);"G "
192 PRINTB; TAB(4);AL$(B); TAB(30);
    PA(B,1);"G "
194 PRINT:PRINT"[DOWN]VUOI MODIFICA
    RE QUALCHE PORZIONE ";
196 INPUT A$:IF A$="NO" THEN 208
198 GOSUB 390
200 GOTO 190
202 REM ***************************
204 REM * VALORI PRINCIPI NUTR.    *
206 REM ***************************
208 PRINT
210 FOR I=1 TO 19:P1$(I)=STR$(PA(A,
    I)):P2$(I)=STR$(PA(B,I)):NEXTI:
    OPZ=1
212 PRINT"[CLEAR]":PRINT"ALIMENTO N
    .";
214 PRINT TAB(17)RV$A; TAB(26)RV$B
216 GOSUB 436
218 PRINT"[DOWN]PREMI UN TASTO PER
    CONTINUARE"
220 GET A$:IF A$="" THEN 220
222 GOTO 162
224 REM ***************************
226 REM * CONFRONTO CALORIE        *
228 REM ***************************
230 PRINT"[CLEAR]":PRINT"[2 DOWN]IN
    TRODUCI LE CALORIE CHE HAI BISO
    GNO"
232 PRINT"[DOWN]OGNI GIORNO, OPPURE
     MOLTIPLICA IL TUO "
234 PRINT"[DOWN]PESO PER 40 :";:INP
    UT CA
235 REM ***************************
236 REM * MODIFICA QUANTITA' DEI   *
238 REM * PRINCIPI NUTRITIVI A     *
240 REM * SECONDA DELLE CAL.       *
242 REM ***************************
246 V(2)=CA:V(3)=(CA*.15)/4
248 V(4)=(CA*.55)/4:V(5)=5
250 V(6)=(CA*.30)/9:V(7)=.36*CA(6)
252 V(8)=1000:V(10)=1000
254 V(11)=400:V(12)=1000
256 V(13)=18:V(14)=1500
258 V(15)=1700:V(16)=20
260 V(17)=60:V(18)=5000:V(19)=400
262 FOR I=1 TO 19:P1$(I)=STR$(V(I))
    :NEXTI:OPZ=0
264 PRINT"[DOWN]VUOI L'ELENCO DEI P
```

```
    RINCIPI NUTRITIVI "
266 PRINT"[DOWN]CONTENUTI IN QUESTA
    QUANTITA'"
268 PRINT"[DOWN]DI CALORIE ";:INPUT
    R$
270 IF R$="NO" THEN 288
272 PRINT"[CLEAR]"
274 PRINT"PRINCIPI NUTRITIVI PER [R
    VS]"CA"[RVOFF] CALORIE"
276 GOSUB 436
278 PRINT"[DOWN]PREMI UN TASTO PER
    CONTINUARE"
280 GET A$:IF A$="" THEN 280
282 REM ***************************
284 REM * INSERIMENTO ALIMENTI     *
286 REM ***************************
288 PRINT"[CLEAR][3 DOWN]VUOI DARE
    UNA LISTA "
290 PRINT"[DOWN]DI CIBI (MAX=20) ";
    :INPUT R$
292 IF R$="NO" THEN 162
294 PRINT"[DOWN]INTRODUCI IL NUMERO
    DEI CIBI"
296 PRINT"[DOWN]TERMINATO SCRIVI ZE
    RO":PRINT
298 FOR I=1 TO 20
300 INPUT SC(I)
302 IF SC(I)=0 THEN 312
304 NEXTI
306 REM ***************************
308 REM * PRESENTAZIONE CIBI INS. *
310 REM ***************************
312 PRINT"[CLEAR]"
314 PRINT"N.  ALIMENTO"; TAB(26)"GR
    AMMI CALORIE"
316 FOR J=1 TO I-1
318 PRINTSC(J); TAB(3)AL$(SC(J)); T
    AB(28);PA(SC(J),1); TAB(34);PA(
    SC(J),2)
320 NEXTJ
322 PRINT"[DOWN]VUOI CAMBIARE QUALC
    HE PORZIONE ";:INPUT R$
324 IF R$="NO" THEN 330
326 GOSUB 390
328 GOTO 322
330 PRINT"[DOWN]VUOI RIVEDERE LA LI
    STA ";:INPUT R$
332 IF R$="SI" THEN 312
334 REM ***************************
336 REM * CARICAMENTO DAI DATI     *
338 REM * DEI PRINCIPI NUTRITIVI   *
340 REM ***************************
342 FOR K=1 TO 19
344 FOR J=1 TO I-1
346 PA(0,K)=PA(0,K)+PA(SC(J),K)
348 NEXTJ
350 NEXTK
352 REM ***************************
354 REM *        CONFRONTO         *
356 REM ***************************
358 FOR I=1 TO 19:P2$(I)=STR$(PA(0,
    I)):NEXTI:OPZ=1
360 PRINT"[CLEAR]":PRINT TAB(15)"VA
    LORI"; TAB(24)"VALORI"
362 PRINT TAB(14)"STANDARD"; TAB(25
    )"LISTA"
364 GOSUB 436
366 PRINT"PREMI UN TASTO PER CONTIN
    UARE "
368 GET A$:IF A$="" THEN 368
370 GOTO 162
372 REM ***************************
374 REM * FORMATI DI STAMPA        *
376 REM ***************************
378 J=J+1:IF  OPZ=0 THEN PRINT"....
    .......";P1$(J):RETURN
380 P1$(J)=MID$(P1$(J),1,6):P2$(J)=
    MID$(P2$(J),1,6)
382 PRINT TAB(17)P1$(J); TAB(24)P2$
    (J):RETURN
384 REM ***************************
386 REM * MODIFICA PORZIONE CIBO   *
388 REM ***************************
390 PRINT"[DOWN]NUMERO DELL'ALIMENT
    O "
392 PRINT"[DOWN]DA MODIFICARE ";
394 INPUT C
396 PRINT"[DOWN]MOLTIPLICARE LA QUA
    NTITA'"
398 PRINT"[DOWN]O NUOVO PESO IN GRA
    MMI(M O P)";:INPUT  A$
400 IF A$="P" THEN 404
402 PRINT"[DOWN]PER QUANTO MOLTIPLI
    CO ";:INPUT M:GOTO 406
404 PRINT"[DOWN]QUANTI GRAMMI ";:IN
    PUT G:M=G/PA(C,1)
406 FOR J=1  TO 19:PA(C,J)=PA(C,J)*
    M:NEXTJ
408 RETURN
410 PRINT"[CLEAR]":END
412 PRINT"[CLEAR]":FOR A=0 TO (CC/1
    2)
414 FOR B=1 TO 12:S1=B+12*A
416 PRINT"[RVS]"S1"[RVOFF]";" .   ";
```

```
    AL$(S1)
418 NEXTB
420 REM ****************************
422 REM * CONTROLLA ULTIMO ALIMENTO
424 REM ****************************
426 IF AL$(S1)="XXXXXXXXX" THEN A=
    (CC/12):GOTO 428
428 PRINT"PREMI UN TASTO PER CONTIN
    UARE"
430 GET A$:IF A$="" THEN 430
432 PRINT"[CLEAR]"
434 NEXTA:GOTO 162
436 J=1
438 REM ****************************
440 REM * PRINCIPI NUTRITIVI       *
442 REM ****************************
444 PRINT:PRINT"CALORIE       ";:GOS
    UB 378
446 PRINT"PROTEINE(G)   ";:GOSUB 378
448 PRINT"CARBOIDR. (G)";:GOSUB 378
450 PRINT"FIBRE(G)      ";:GOSUB 378
452 PRINT"GRASSI (G)    ";:GOSUB 378
454 PRINT"POLINS.(G)    ";:GOSUB 378
456 PRINT"SALI(MG)      ";:GOSUB 378
458 PRINT"POTASSIO(MG) ";:GOSUB 378
460 PRINT"CALCIO(MG)    ";:GOSUB 378
462 PRINT"MAGNESIO(MG) ";:GOSUB 378
464 PRINT"FOSFORO(MG)   ";:GOSUB 378
466 PRINT"FERRO(MG)     ";:GOSUB 378
468 PRINT"VIT.B1(MCG)   ";:GOSUB 378
470 PRINT"VIT.B2(MCG)   ";:GOSUB 378
472 PRINT"VIT.PP(MG)    ";:GOSUB 378
474 PRINT"VIT.C(MG)     ";:GOSUB 378
476 PRINT"VIT.A(UI)     ";:GOSUB 378
478 PRINT"VIT.D(UI)     ";:GOSUB 378
480 RETURN
482 REM ****************************
484 REM * DATI DEGLI ALIMENTI      *
486 REM ****************************
488 DATA "TAZZA DI LATTE INTERO",24
    4,159
490 DATA 8.5,12,0,8.6,0,122,352,288
    ,32,227,0,70,420,.2,2,340,0
492 DATA "TAZZA DI PISELLI",110,29
494 DATA 2.7,4.9,.8,.6,0,25,234,152
    ,57,39,.6,140,200,1.2,46,5400,0
496 DATA "TAZZA DI FUNGHI",100,17
498 DATA 1.9,2.4,.6,.1,0,400,97,6,1
    3,68,.5,20,250,2,2,0,0
500 DATA "UN POMODORO",100,22
502 DATA 1.1,4.7,.5,.2,0,3,244,13,1
    4,27,.5,60,40,.7,23,900,0
504 DATA "UNA CIPOLLA",50,22
506 DATA 1,45,.5,0,0,5,80,15,5,20,.
    5,15,20,.1,5,20,0
508 DATA "1 TAZZA DI FAGIOLI",100,2
    8
510 DATA 3.2,5.2,.7,.2,0,4,256,17,0
    ,48,.9,90,100,.7,6,20,0
512 DATA "CUCCHIAIO DI SOTTOACETI",
    30,3
514 DATA .2,.7,.1,.1,0,0,0,8,0,6,.4
    .0,18,0,2,93,0
516 DATA "TRANCIO DI FOCACCIA",100,
    340
518 DATA 5,72,0,18,0,0,0,0,0,0,0,0,
    0,0,0,0,0
520 DATA "MELONE",150,45
522 DATA 1,11.2,.4,.2,0,18,376,21,2
    4,24,.6,60,45,.9,50,5100,0
524 DATA "SUCCO DI POMPELMO",100,43
526 DATA .6,10.1,.1,.2,0,1,184,10,1
    2,15,.3,50,20,.2,34,100,0
528 DATA "SUCCO D'ARANCIA",100,45
530 DATA .7,10.4,.1,.2,0,1,200,11,1
    1,17,.2,90,30,.4,50,200,0
532 DATA "RISO BOLLITO",100,109
534 DATA 2,24.2,.1,.1,0,374,28,10,8
    ,28,.9,110,0,1,0,0,0
536 DATA "UN PANINO BIANCO",23,62
538 DATA 2,11.6,0,.7,0,117,24,19,5,
    22,.6,60,50,.6,0,0,0
540 DATA "CASTAGNE",100,198
542 DATA 3,42,10,2,3,1,90,30,0,0,.7
    ,0,0,1.2,0,0,0
544 DATA "BRACIOLA DI MAIALE",100,4
    00
546 DATA 23,0,0,30,2.5,65,400,14,27
    ,270,3,900,280,6,0,0,0
548 DATA "PETTO DI POLLO ARROSTO",1
    00,140
550 DATA 24,0,0,4,1,65,275,10,22,20
    0,17,50,190,8,0,90,0
552 DATA "TROTA BOLLITA",100,88
554 DATA 14,0,0,4,0,80,300,10,16,15
    0,.4,30,40,5,0,400,0
556 DATA "CUCCCHIAIO DI MAIONESE",1
    4,101
558 DATA .15,.3,0,11.2,0,84,5,3,0,4
    ,.01,3,6,0,0,39,0
560 DATA "UN UOVO",54,88
562 DATA 7,.5,0,6.2,.5,66,70,29,6,1
    10,1.2,60,160,.1,0,640,27
564 DATA "XXXXXXXXX"
```

ALTA MATEMATICA

di **Eugenio Coppari**

Tutto sui sistemi lineari, nonché la risoluzione delle equazioni di secondo terzo e quarto grado sia nel campo reale che nel campo complesso.

La teoria delle equazioni è da sempre uno dei campi più fertili della ricerca matematica. Sin dal tempo dei Babilonesi è nota la formula che consente la risoluzione delle equazioni di secondo grado: essa richiede l'estrazione della radice quadrata da una relazione che lega vicendevolmente i coefficenti (B elevato al quadrato meno 4AC). La scuola matematica francese è senza dubbio una delle più prolifiche per quanto concerne la ricerca in questo campo teorico. È quindi opportuno ricordare brevemente le vicissitudini di un noto matematico del secolo scorso: Evariste Galois.

Galois che, pur essendo morto in giovanissima età (20 anni), ha consentito con i propri lavori un notevole progresso di questo settore importantissimo dell'algebra. Sino alla metà dell'ottocento tutti i maggiori matematici erano soliti porsi un problema fondamentale: Sotto quali condizioni è possibile verificare la risolubilità di una equazione?

Cerchiamo di approfondire, per quanto possibile, il significato di una tale affermazione. Se vogliamo essere maggiormente precisi nella formulazione dell'interrogativo, sarà necessario enunciarlo in modo notevolment più dettagliato: Esiste un metodo generalizzato che consenta la risoluzione di una equazione comprendente la sola variabile X e avente dei coefficenti appartenenti all'insieme dei numeri naturali?

Diamo un semplice esempio di questo interrogativo: l'equazione X elevato alla sesta, più 4 per X elevato alla seconda uguale a zero, risponde assai facilmente alle caratteristiche richieste in precedenza. Essa contiene effettivamente la sola variabile X e i due coefficenti appartengono entrambi ai numeri naturali. Anche se appuriamo che questa equazione, e molte altre ancora, sono risolubili, ciò non può rappresentare una garanzia per quanto concerne le possibilità di scoperta di un algoritmo generalizzato.

È proprio partendo da questa considerazione che si può meglio comprendere la rivelanza del lavoro di Galois. Egli riuscì a dimostrare tramite la teoria dei gruppi un asserto di fondamentale rilevanza: "per ogni equazione di grado superiore o uguale al quinto non esiste un metodo generale che consenta di appurarne la sua risolubilità".

Certamente possono esistere casi particolari in merito ai quali è immediatamente chiaro il risultato, ad esempio X elevato alla nona uguale a due, probabilmente non creerebbe dei problemi neanche ad un ragazzo delle medie inferiori, ma questo non è assolutamente di alcuna rilevanza nei confronti del teorema enunciato e dimostrato da Galois.

Questi concetti iniziali delimitano chiaramente i limiti teorici entro i quali potrà agire il programma che verrà ora analizzato.

Questo software consente il calcolo di equazioni di secondo, terzo, quarto grado e sistemi lineari comprendenti tre variabili. Le righe che sono situate tra **1200** e **1260** rendono possibile la creazione della mascherina del programma. In essa sono contenute alcune POKE che fanno riferimento ad aree assai importanti del computer. Ma poiché il loro utilizzo in questo caso è finalizzato semplicemente a un fattore di carattere estetico possono essere agevolmente e senza titubanze rimosse da chi voglia una estetica diversa da quella da me proposta. Nelle linee da **1260** sino a **1740** è situata la zona di programma che consente di visualizzare i calcoli compiuti dal vostro Commodore.

Sono naturalmente state contemplate tutte le possibilità che possono avere luogo durante la fase di output, trattandosi di un programma che può analizzare tutti e tre i casi di potenza (seconda, terza e quarta). Inoltre il risultato, in caso di bisogno, verrà anche analizzato al di fuori dell'usuale campo dei numeri reali cercando il risultato immaginario dell'equazione. Naturalmente in questa evenienza verrà visualizzata la lettera "I" che all'interno del campo complesso, se elevata ad una potenza di ordine pari, ha un valore negativo (-1).

Nelle righe a partire dalla **2360** viene calcolata l'equazione di quarto grado. Dalla **2960** quella di terzo grado ed infine dalla linea **3500** quella di secondo grado. Analizziamo in maniera particolareggiata i tre diversi algoritmi.

Equazioni di secondo grado

Le possibilità dell'equazione A moltiplicato per X elevato al quadrato più B moltiplicato per X più C, posto "q" uguale alla radice di B elevato al quadrato meno 4 per C, sono le seguenti.

- se q = >O allora esistono 2 radici reali uguali o distinte.
- se q = <O allora esistono due radici complesse.

Equazioni di terzo grado

Immaginiamo di avere l'equazione X elevato al cubo più A2 moltiplicato per X elevato al quadrato più A1 moltiplicato per X più AO uguale a O, a questo punto si può facilmente dimostrare che esiste una radice negativa nell'intervallo -(1 + v), O se AO>O e una positiva nell'intervallo 0, (1 + v) se AO<O.

In questo caso "v" è il massimo dei valori assoluti (pro-

prio come l'effetto dell'istruzione BASIC ABS) fra i coefficienti della equazione in esame.

Naturalmente si verificherà uno solo dei due casi e questo ci permetterà di trovare il primo dei tre possibili valori. Per quanto concerne gli altri due valori si procederà con il metodo che prende il nome detto di "bisezione dell'intervallo". Come possiamo spiegare in termini semplici quest'ultimo concetto? Prendiamo ad esempio il sistema della ricerca dicotomica che provvede a analizzare una serie di dati partendo dai lati estremi procedendo verso il centro fino a trovare ciò che si cerca; prendo il punto centrale di un insieme e vedo da che parte debbo andare, in basso od in alto, e così procedo fino alla risoluzione della ricerca. Così la bisezione dell'intervallo esamina una porzione di valori sempre più piccola e tendente alla perfezione del numero da cercare.

Equazioni di quarto grado

Estremamente banale il caso della equazione di quarto grado che può essere ricondotto ad una eventualità precedente scomponendo l'equazione originale in due di secondo grado.

Conclusione

Al termine di questa parte di programma concernente la risoluzione di equazioni è posta la routine di input controllato che non rende possibile l'infiltrazione, in fase di digitazione, dei dati di elementi estranei.

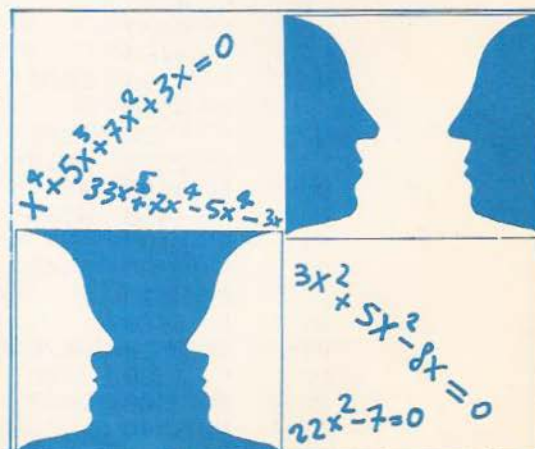
Nella seconda parte del programma è affrontato il tema dei sistemi lineari, nel caso specifico si utilizzeranno 3 equazioni con conseguenti tre incognite.

L'argomento viene affrontato facendo ricorso a strumenti di carattere tipicamente universitario: la regola di Cramer (quella che ci permette tramite il calcolo matriciale di risolvere sistemi di tre equazioni lineari, ne una di più ne una di meno) consente un calcolo dei valori da ricercare, se possibile, in tempi particolarmente ristretti. Ricordo quindi a chi sia particolarmente interessato all'argomento, che esso è affrontato usualmente su qualsiasi testo di geometria delle facoltà universitarie.

```
C R O S S   R E F E R E N C E

-------*------------------- - - -
VAR.   ! LINEA DEL PROGRAMMA
-------*------------------- - - -
       !
A      !  2360  2380  2400  2420  3080  3100
       !  3140  3160  3200  4040  4140  4260
       !  4384  4400  4440  4460
A(     !  1540  1620  2220  2360  2440  2460
       !  2480  2500  2520  2540  2780  2800
       !  2840  2860  2900  2960  2980  3020
       !  3080  3120  3400  3420  3500  3520
A1     !  4060  4180  4260  4280  4384  4400
       !  4440  4460
A2     !  4080  4220  4260  4384  4440
AM     !  2980  3040  3080  3100
B      !  2360  2380  2400  2420  3080  3100
       !  3140  3160  3220  4040  4140  4260
       !  4280  4320  4340  4440  4460
B1     !  4060  4180  4260  4320  4440
B2     !  4080  4220  4260  4320  4340  4440
       !  4460
C      !  2360  2400  2420  4040  4140  4260
       !  4320  4384
C1     !  4060  4180  4260  4320  4340  4384
       !  4400
C2     !  4080  4220  4260  4280  4320  4340
       !  4384  4400
D      !  2360  2420  4040  4140  4320  4340
       !  4384  4400  4440
```

```
D1     !  4060  4180  4320  4340  4384  4440
       !  4460
D2     !  4080  4220  4320  4384  4400  4440
       !  4460
E      !  3180  4260  4280
E1     !  4320  4340
E2     !  4384  4400
E3     !  4440  4460
ER     !  3180  3300
F      !  4260  4280
F(     !  3120  3160  3260  3320
F1     !  4320  4340
F2     !  4384  4400
F3     !  4440  4460
G      !  4260  4280
G1     !  4320  4340
G2     !  4384  4400
G3     !  4440  4460
H      !  4260  4280
H1     !  4320  4340
H2     !  4384  4400
H3     !  4440  4460
I      !  1580  1600  1620  1640  2200  2220
       !  2240  3000  3020  3060  4260  4280
I1     !  4340
I2     !  4400
I3     !  4460
II     !  2080  2100  2120  2140  2840
IM     !  1780  1800  1820  1840  1900  1920
       !  1940  1960  2000  2020  2040  2060
       !  2840  3600  3700  3780
IN$    !  1620  4699  4800  4880  4900
K      !  2680  2700  2740  2780  2860
L      !  2760  2800  4280
L1     !  4340
L2     !  4400
L3     !  4460
M      !  2760  2860  4280  4500  4520  4540
       !  4560
N      !  1320  1360  1580  1620  1740  2200
       !  2220  2260  4340  4520
O      !  4400  4540
P      !  4460  4560
PI     !  4100
Q      !  2380  2440  2460  2700  2720  2760
       !  4520
Q(     !  3680
Q1     !  3520  3620  3800  3820
Q2     !  3580  3600  3760  3800  3820
R      !  2400  2480  2500  2700  2740  2760
       !  4540
```

```
R(    !  1540  1780  1800  1820  1840  1880
      !  1900  1920  1940  1960  2000  2020
      !  2040  2060  2080  2100  2120  2140
      !  2580  2680  2720  2840  2900  3360
      !  3620  3680  3800  3820
RH    !  4699  4820  4860
RY$   !  3880
S     !  2420  2460  2700  4560
SW    !  2640  3380
SZ$   !  1200  1240  1260
T     !  1300  3920
TE    !  3260  3280  3320
TI    !  4699  4720
TT    !  3020  3040
UT    ! 10000
W     !  3500  3540  3580  3760
X     !  3120  4520  4600
XA    !  3140  3240  3300  3320
XB    !  3140  3240  3300  3340
XC    !  2960  3200  3220  3240  3260  3320
      !  3340  3360  3400  3420
Y     !  4540  4620
Y1    !  3160  3200
Y2    !  3160  3220
YQ    !  1360
YT    !  1660
Z     !  4560  4640  4760  4780  4800  4820
      !  4840  4860  4900
Z$    !  4700  4760  4880
ZC    !  4699  4720
ZT    !  4699  4720
```

```
100 REM  *************************
110 REM  *                       *
120 REM  *     RADICI  DELLE     *
130 REM  *                       *
140 REM  *       EQUAZIONI       *
150 REM  *                       *
160 REM  *  E SISTEMI LINEARI    *
170 REM  *                       *
180 REM  *************************
190 REM  *                       *
200 REM  *    AUTHOR SOFTWARE    *
210 REM  *                       *
220 REM  *    EUGENIO COPPARI    *
230 REM  *                       *
240 REM  *************************
250 REM  *                       *
260 REM  *   VC 20  (16K)    SI  *
```

```
270 REM  *    COMMODORE 64      SI   *
280 REM  *    COMMODORE 4000    SI   *
290 REM  *    COMMODORE 8000    SI   *
300 REM  *    COMMODORE 16      SI   *
310 REM  *    COMM.PLUS 4       SI   *
320 REM  ************************
340 REM
500 REM
600 REM  ************************
610 REM  *                      *
620 REM  *     COPERTINA  DI     *
630 REM  *     PRESENTAZIONE     *
640 REM  *         DEL           *
650 REM  *      PROGRAMMA        *
660 REM  *                      *
670 REM  ************************
1000 POKE 53230,238:POKE 53231,32:PO
     KE 53232,208:POKE 53233,76:POKE
      53234,49
1020 POKE 53235,234:POKE 56333,127:P
     OKE 788,238:POKE 789,207:POKE 5
     6333,129
1040 POKE 56325,1:POKE 676,1
1060 PRINTCHR$(147):POKE 53281,2
1080 PRINT"[7 DOWN][6 RIGHT][RVS]
                              "
1100 PRINT"[6 RIGHT][RVS] [BIANCO]A[
     CELESTE][RVOFF][RVS] EQUAZIONI
     DI 2,3,4 GRADO "
1120 PRINT"[6 RIGHT][RVS]
                        [RVOFF]"
1140 PRINT"[3 DOWN][6 RIGHT][RVS]
                              "
1160 PRINT"[6 RIGHT][RVS] [BIANCO]B[
     CELESTE][RVOFF][RVS]   SISTEMI
      LINEARI      "
1180 PRINT"[6 RIGHT][RVS]
                        [RVOFF]"
1200 GET SZ$:IF SZ$<>"A" AND SZ$<>"B
     " THEN 1200
1220 POKE 56333,127:POKE 789,234:POK
     E 788,49:POKE 56333,129:POKE 56
     325,21
1240 IF  SZ$="A" THEN 1280
1260 IF  SZ$="B" THEN 3920
1261 REM  ************************
1262 REM  *                      *
1263 REM  *    CALCOLO DELLE     *
1264 REM  *       RADICI         *
1265 REM  *      DI  UNA         *
1266 REM  *     EQUAZIONE        *
1267 REM  *                      *
1268 REM  ************************
1280 PRINT"[CLEAR]"
1300 POKE 53280,6:POKE 53281,6:T=1
1320 INPUT "[RVS]QUAL'E' IL GRADO DE
     LLA EQUAZIONE[RVOFF]  [RVS]";N:
     PRINT
1340 PRINT"[NERO]┌──────────────────
     ─────────┐
1360 FOR YQ=1 TO N:PRINT"|
                 |"
1380 PRINT"|
                 |"
1400 PRINT"|
                 |"
1420 PRINT"├──────────────────
     ─────┤
1440 NEXT
1460 PRINT"|
                 |"
1480 PRINT"|
                 |"
1500 PRINT"|
                 |"
1520 PRINT"└──────────────────
     ─────┘[CELESTE]"
1540 DIM A(5,5),R(4)
1560 PRINT"[HOME][DOWN]"
1580 FOR I=N TO 0 STEP -1
1600 PRINT"[3 DOWN][RIGHT] [RVS]COEF
     FICENTE DI X↑ [LEFT]";I;
1620 GOSUB 4680:A(N,I)=VAL(IN$):PRIN
     T
1640 NEXTI
1660 FOR YT=1 TO 1000:NEXTYT
1680 PRINT"[CLEAR]"
1700 PRINT:PRINT
1720 GOSUB 2180
1740 ON N-1GOTO 1760,1860,1980
1760 REM  ************************
1761 REM  *                      *
1762 REM  *   GESTIONE FINALE    *
1763 REM  *    DEI RISULTATI     *
1764 REM  *       OTTENUTI       *
1765 REM  *                      *
1766 REM  ************************
1780 IF IM=0 THEN PRINT"VALORE DELLA
      PRIMA RADICE = [RVS]";R(1):PRI
     NT"[RVOFF]"
1800 IF IM=0 THEN PRINT"VALORE DELLA
      SECONDA RADICE = [RVS]";R(2):P
     RINT"[RVOFF]":GOTO 3860
1820 PRINT"VALORE DELLA PRIMA RADICE
```

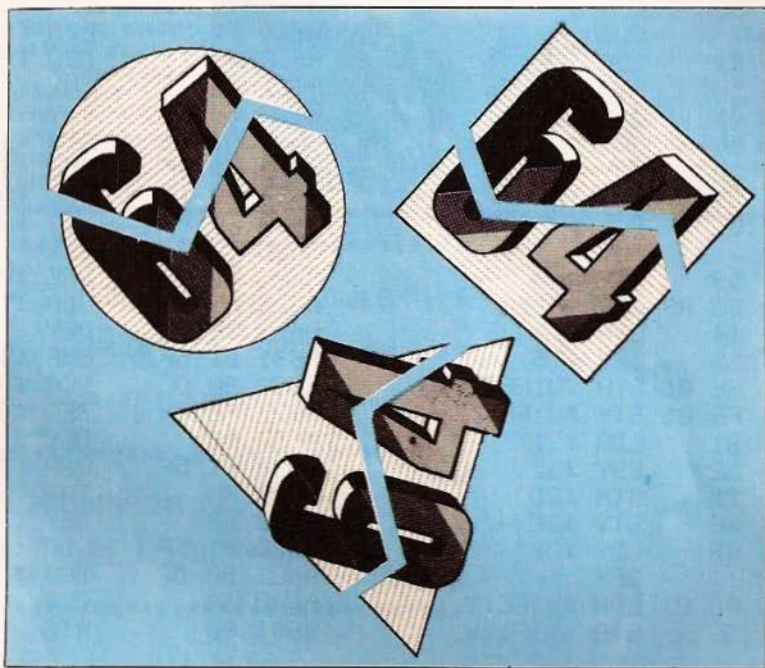
```
 = [RVS]";R(1);"   +I  ";IM:PRI
 NT"[RVOFF]"
1840 PRINT"VALORE DELLA SECONDA RADI
 CE = [RVS]";R(2);"   -I  ";IM:P
 RINT"[RVOFF]":GOTO 3860
1860 REM
1880 PRINT"VALORE DELLA PRIMA RADICE
 = [RVS]";R(3):PRINT"[RVOFF]"
1900 IF  IM =0  THEN  PRINT"VALORE D
 ELLA SECONDA RADICE = [RVS]";R(
 1):PRINT"[RVOFF]"
1920 IF  IM=0 THEN PRINT"VALORE DELL
 A TERZA RADICE = [RVS]";R(2):PR
 INT"[RVOFF]":GOTO 3860
1940 PRINT"VALORE DELLA SECONDA RADI
 CE = [RVS]";R(1);"   +I  ";IM:P
 RINT"[RVOFF]"
1960 PRINT"VALORE DELLA TERZA RADICE
 = [RVS]";R(2);"   -I  ";IM:PR
 INT"[RVOFF]":GOTO 3860
1980 REM
2000 IF  IM=0 THEN PRINT"VALORE DELL
 A PRIMA RADICE = [RVS]";R(1):PR
 INT"[RVOFF]"
2020 IF  IM=0 THEN PRINT"VALORE DELL
 A SECONDA RADICE = [RVS]";R(2):
 GOTO 2080:PRINT"[RVOFF]"
2040 PRINT"VALORE DELLA PRIMA RADICE
 = [RVS]";R(1);"   +I  ";IM:PRI
 NT"[RVOFF]"
2060 PRINT"VALORE DELLA SECONDA RADI
 CE = [RVS]";R(2);"   -I  ";IM:P
 RINT"[RVOFF]"
2080 IF II=0 THEN PRINT"VALORE DELLA
  TERZA RADICE = [RVS]";R(3):PRI
 NT"[RVOFF]"
2100 IF II=0 THEN  PRINT"VALORE DELL
 A QUARTA RADICE = [RVS]";R(4):P
 RINT"[RVOFF]":CLR :GOTO 3860
2120 PRINT"VALORE DELLA TERZA RADICE
 = [RVS]";R(3);"   +I  ";II:PRI
 NT"[RVOFF]"
2140 PRINT"VALORE DELLA QUARTA RADIC
 E = [RVS]";R(4);"   -I  ";II:PR
 INT"[RVOFF]"
2160 GOTO 3860
2161 REM  *************************
2162 REM  *                       *
2163 REM  *   SALTI ALLE VARIE    *
2164 REM  *       ROUTINE         *
2165 REM  *     CHE CALCO-        *
2166 REM  *    LANO LE RADICI     *
2167 REM  *************************
2180 REM
2200 FOR I=0 TO N-1
2220 A(N,I)=A(N,I)/A(N,N)
2240 NEXTI
2260 ON N-1GOTO 2280,2320,2360
2280 GOSUB 3480
2300 RETURN
2320 GOSUB 2940
2340 RETURN
2350 REM *************************
2351 REM *                       *
2352 REM *    CALCOLO DELLA      *
2353 REM *    EQUAZIONE DI       *
2354 REM *    QUARTO  GRADO      *
2355 REM *                       *
2356 REM *************************
2360 A=A(4,3):B=A(4,2):C=A(4,1):D=A(
 4,0)
2380 Q=B-(3*A*A/8)
2400 R=C-(A*B/2)+(A*A*A/8)
2420 S=D-(A*C/4)+(A*A*B/16)-(3*A*A*A
 *A/256)
2440 A(3,2)=Q/2
2460 A(3,1)=(Q*Q-4*S)/16
2480 A(3,0)=-(R*R/64)
2500 IF (R<>0) OR (A(3,1)>=0) THEN 2
 620
2520 A(2,1)=A(3,2)
2540 A(2,0)=A(3,1)
2560 GOSUB 3480
2580 R(3)=R(1)
2600 GOTO 2680
2620 REM
2640 SW=-1
2660 GOSUB 2940
2680 K=SQR(R(3))
2700 IF K=0 THEN R=SQR(Q*Q-4*S):GOTO
  2760
2720 Q=Q+(4*R(3))
2740 R=R/(2*K)
2760 L=(Q-R)/2:M=(Q+R)/2
2780 A(2,1)=2*K
2800 A(2,0)=L
2820 GOSUB 3500
2840 R(3)=R(1)-(A(4,3)/4):R(4)=R(2)-
 (A(4,3)/4):II=IM
2860 A(2,1)=-2*K:A(2,0)=M
2880 GOSUB 3500
2900 R(2)=R(2)-(A(4,3)/4):R(1)=R(1)-
 (A(4,3)/4)
2920 RETURN
```

```
2940 REM
2941 REM  ************************
2942 REM  *                      *
2943 REM  *    CALCOLO DELLE     *
2944 REM  *        RADICI        *
2945 REM  *   DELLE EQUAZIONI    *
2946 REM  *   DI  TERZO GRADO    *
2947 REM  *                      *
2948 REM  ************************
2960 IF A(3,0)=0 THEN XC=0:GOTO 3360
2980 AM=ABS(A(3,0))
3000 FOR I=1 TO 3
3020 TT=ABS(A(3,I))
3040 IF (AM<TT) THEN AM=TT
3060 NEXTI
3080 IF A(3,0)>0 THEN A=-AM-1:B=0:GO
     TO 3120
3100 A=0:B=AM+1
3120 DEF FNF(X)=X*X*X+A(3,2)*X*X+A(3
     ,1)*X+A(3,0)
3140 XA=A:XB=B
3160 Y1=FNF(A):Y2=FNF(B)
3180 ER=1E-6
3200 IF Y1=0 THEN XC=A:GOTO 3360
3220 IF Y2=0 THEN XC=B:GOTO 3360
3240 XC=(XA+XB)/2
3260 TE=FNF(XC)
3280 IF TE=0 THEN GOTO 3360
3300 IF (XB-XA)<ER THEN GOTO 3360
3320 IF (FNF(XA)*TE)>0 THEN XA=XC:GO
     TO 3240
3340 XB=XC:GOTO 3240
3360 R(3)=XC
3380 IF SW=-1 THEN RETURN
3400 A(2,1)=A(3,2)+XC
3420 A(2,0)=A(3,1)+A(3,2)*XC+XC*XC
3440 GOSUB 3480
3460 RETURN
3480 REM
3481 REM  ************************
3482 REM  *                      *
3483 REM  *    CALCOLO DELLE     *
3484 REM  *    RADICI  DELLE     *
3485 REM  *    EQUAZIONI  DI     *
3486 REM  *    SECONDO GRADO     *
3487 REM  *                      *
3488 REM  ************************
3500 W=A(2,1)*A(2,1)-4*A(2,0)
3520 Q1=-A(2,1)/2
3540 ON ( SGN(W)+2)GOTO 3560,3660,37
     40
3560 REM
3580 Q2=SQR(-W)/2
3581 REM  ************************
3582 REM  *                      *
3583 REM  *    CALCOLI PER LA    *
3584 REM  *   EVENTUALE PARTE    *
3585 REM  *     IMMAGINARIA      *
3586 REM  *                      *
3587 REM  ************************
3600 IM=Q2
3620 R(1)=Q1:R(2)=Q1
3640 RETURN
3660 REM
3680 R(1)=Q(1):R(2)=Q(1)
3700 IM=0
3720 RETURN
3740 REM
3760 Q2=SQR(W)/2
3780 IM=0
3800 R(1)=Q1+Q2
3820 R(2)=Q1-Q2
3840 RETURN
3841 REM  ************************
3842 REM  *                      *
3843 REM  *  FINE PRG EQUAZIONI  *
3844 REM  *                      *
3845 REM  ************************
3846 REM  ************************
3847 REM  *                      *
3848 REM  *    CALCOLO SISTE-    *
3849 REM  *     MI LINEARI       *
3850 REM  *                      *
3851 REM  ************************
3860 PRINT"[HOME][18 DOWN][4 RIGHT]P
     REMERE UN TASTO PER CONTINUARE"
3880 GET RY$:IF RY$="" THEN 3880
3900  RUN
3920 POKE 53280,6:POKE 53281,6:PRINT
     "[BIANCO]":T=1
3940 PRINT"[CLEAR]  SISTEMI A TRE IN
     COGNITE X;Y;Z";"[CELESTE]"
3960 PRINT"[DOWN]1" EQUAZIONE IN ESA
     ME  : A[NERO]X[CELESTE]+B[NERO]
     Y[CELESTE]+C[NERO]Z[CELESTE]=D"
3980 PRINT"[DOWN]2" EQUAZIONE IN ESA
     ME  : A1[NERO]X[CELESTE]+B1[NER
     O]Y[CELESTE]+C1[NERO]Z[CELESTE]
     =D1"
4000 PRINT"[DOWN]3" EQUAZIONE IN ESA
     ME  : A2[NERO]X[CELESTE]+B2[NER
     O]Y[CELESTE]+C2[NERO]Z[CELESTE]
     =D2"
4020 PRINT"[DOWN]FORNITE I PARAMETRI
```

```
      RICHIESTI"
4021 REM  ************************
4022 REM  *                      *
4023 REM  *    VIENE UTILIZZATO  *
4024 REM  *    PER IL CALCOLO    *
4025 REM  *    IL   METODO   DI  *
4026 REM  *       DI  CRAMER     *
4027 REM  *                      *
4028 REM  ************************
4040 INPUT "[DOWN] (A,B,C,D) :";A,B,
     C,D
4060 INPUT "[DOWN] (A1,B1,C1,D1) :";
     A1,B1,C1,D1
4080 INPUT "[DOWN] (A2,B2,C2,D2) :";
     A2,B2,C2,D2
4100 FOR PI=1 TO 5000:NEXT
4120 PRINT"[CLEAR][DOWN]   IL SISTEM
     A :"
4140 PRINT"[DOWN] / ";A;"[LEFT][NERO]
     X[CELESTE] +";B;"[LEFT][NERO]Y[
     CELESTE] +";C;"[LEFT][NERO]Z[CE
     LESTE] ="D
4160 PRINT" !"
4180 PRINT"{ ";A1;"[LEFT][NERO]X[CEL
     ESTE] +";B1;"[LEFT][NERO]Y[CELE
     STE] +";C1;"[LEFT][NERO]Z[CELES
     TE] =";D1
4200 PRINT" !"
4220 PRINT"\ ";A2;"[LEFT][NERO]X[CEL
     ESTE] +";B2;"[LEFT][NERO]Y[CELE
     STE] +";C2;"[LEFT][NERO]Z[CELES
     TE] =";D2
4240 REM  ************************
4241 REM  *                      *
4242 REM  *  CALCOLO DI DELTA X  *
4243 REM  *                      *
4244 REM  ************************
4260 E=(A*B1*C2):F=(B*C1*A2):G=(C*A1
     *B2):H=(A2*B1*C):I=(B2*C1*A)
4280 L=(C2*A1*B):M=E+F+G-H-I-L
4300 REM  CALCOLO DI DELTA X (N)
4320 E1=(D*B1*C2):F1=(B*C1*D2):G1=(C
     *D1*B2):H1=(D2*B1*C)
4340 I1=(B2*C1*D):L1=(C2*D1*B):N=E1+
     F1+G1-H1-I1-L1
4360 REM  ************************
4370 REM  *                      *
4380 REM  *  CALCOLO DI DELTA Y  *
4381 REM  *                      *
4382 REM  ************************
4384 E2=(A*D1*C2):F2=(D*C1*A2):G2=(C
     *A1*D2):H2=(A2*D1*C)
4400 I2=(D2*C1*A):L2=(C2*A1*D):O=E2+
     F2+G2-H2-I2-L2
4420 REM  ************************
4421 REM  *                      *
4422 REM  *  CALCOLO DI DELTA Z  *
4423 REM  *                      *
4424 REM  ************************
4440 E3=(A*B1*D2):F3=(B*D1*A2):G3=(D
     *A1*B2):H3=(A2*B1*D)
4460 I3=(B2*D1*A):L3=(D2*A1*B):P=E3+
     F3+G3-H3-I3-L3
4480 REM  CALCOLO INCOGNITE X,Y,Z
4500 IF  M=0 THEN PRINT"[CLEAR][RVS]
     [7 DOWN][4 RIGHT]IL SISTEMA NON
      AMMETTE SOLUZIONI[RVOFF]":GOTO
      10000
4520 Q=N/M:X=(Q)
4540 R=O/M:Y=(R)
4560 S=P/M:Z=(S)
4580 PRINT"[2 DOWN]  HA SOLUZIONI  "
4600 PRINT"[DOWN]    [NERO]X[CELESTE]
     =";X
4620 PRINT"[DOWN]    [NERO]Y[CELESTE]
     =";Y
4640 PRINT"[DOWN]    [NERO]Z[CELESTE]
     =";Z
4660 GOTO 3860
4670 REM  ************************
4680 REM  *                      *
4690 REM  *   ROUTINE DI INPUT   *
4691 REM  *     CONTROLLATO      *
4692 REM  *                      *
4693 REM  ************************
4699 IN$=" ":ZT=TI:ZC=2:RH=0
4700 GET Z$:IF  Z$<>"" THEN 4760
4720 IF ZT<=TI THEN PRINTMID$(" ▒",Z
     C,1);"[LEFT]";:ZC=3-ZC:ZT=TI+15
4740 GOTO 4700
4760 Z=ASC(Z$)
4780 IF Z=13 THEN 4900
4800 IF Z=45 AND IN$=" " THEN 4880
4820 IF Z=46 AND RH<>1 THEN 4860
4840 IF Z<48 OR Z>57 THEN 4700
4860 IF Z=46 THEN RH=1
4880 IN$=IN$+Z$:IF Z$<>"▒" THEN PRIN
     TZ$;
4900 IF Z=13 OR LEN(IN$)=8 THEN IN$=
     MID$(IN$,2):PRINT"[BLEU] [CELES
     TE]";:RETURN
4920 GOTO 4700
10000 FOR UT=1 TO 3000:NEXTUT:GOTO 38
      60
```

IL MERGE E L'APPEND

Le Routine in L.M. per tutti i tipi di Commodore

di Giancarlo de Cobelli

Dopo aver presentato sul numero scorso una routine in L.M. che permetteva di creare l'autostart ad un qualsiasi programma tramite loader vi.propongo su questo numero altre due routine di L.M. molto utili che non sono presenti nel BASIC dei Commodore: il MERGE e l'APPEND.

Le routines in questione permettono di fondere od aggiungere un programma che è residente su disco ad un programma precedentemente caricato nella memoria RAM del computer.

L'utilizzo di queste routine è utile poiché i programmi che si realizzano vengono di solito creati non in una volta sola, ma costruendo tante subroutines a se stanti che vengono appunto fuse o collegate insieme.

Per comodità, le locazioni di memoria delle routines sono riferite al Commodore 64; per i riferimenti agli altri tipi confrontare i disassemblati riportati nelle relative figure.

MERGE

Questa prima routine elimina tutte quelle preoccupazioni da cui i programmatori sono assillati quando devono caricare un programma da disco da collegare o meglio fondere, con un programma residente in memoria. Per realizzare questo bisogna utilizzare alcune subroutine che sono presenti all'interno della ROM BASIC.

Queste routine collegate insieme permettono di scrivere una riga di programma letta da qualsiasi periferica ed inserirla al giusto posto nel programma in memoria secondo il numero di riga. La subroutine è locata all'indirizzo di memoria $A49C; le prime tre istruzioni non ci interessano poiché abbiamo già in memoria la linea da inserire (vedremo dopo come) e quindi il programma sal-

```
A49C 20 6B A9 JSR $A96B
A49F 20 79 A5 JSR $A579
A4A2 84 0B    STY $0B
A4A4 20 13 A6 JSR $A613
A4A7 90 44    BCC $A4ED


A4ED 20 59 A6 JSR $A659
A4F0 20 33 A5 JSR $A533
A4F3 AD 00 02 LDA $0200
A4F6 F0 88    BEQ $A480
A4F8 18       CLC
A4F9 A5 2D    LDA $2D
A4FB 85 5A    STA $5A
A4FD 65 0B    ADC $0B
A4FF 85 58    STA $58
A501 A4 2E    LDY $2E
A503 84 5B    STY $5B
A505 90 01    BCC $A508
A507 C8       INY
A508 84 59    STY $59
A50A 20 B8 A3 JSR $A3B8
A50D A5 14    LDA $14
A50F A4 15    LDY $15
A511 8D FE 01 STA $01FE
A514 8C FF 01 STY $01FF
A517 A5 31    LDA $31
A519 A4 32    LDY $32
A51B 85 2D    STA $2D
A51D 84 2E    STY $2E
A51F A4 0B    LDY $0B
A521 88       DEY
A522 B9 FC 01 LDA $01FC,Y
A525 91 5F    STA ($5F),Y
```

```
A527 88       DEY
A528 10 F8    BPL $A522
A52A 20 59 A6 JSR $A659
A52D 20 33 A5 JSR $A533
A530 4C 80 A4 JMP $A480

A613 A5 2B    LDA $2B
A615 A6 2C    LDX $2C
A617 A0 01    LDY #$01
A619 85 5F    STA $5F
A61B 86 60    STX $60
A61D B1 5F    LDA ($5F),Y
A61F F0 1F    BEQ $A640
A621 C8       INY
A622 C8       INY
A623 A5 15    LDA $15
A625 D1 5F    CMP ($5F),Y
A627 90 18    BCC $A641
A629 F0 03    BEQ $A62E
A62B 88       DEY
A62C D0 09    BNE $A637
A62E A5 14    LDA $14
A630 88       DEY
A631 D1 5F    CMP ($5F),Y
A633 90 0C    BCC $A641
A635 F0 0A    BEQ $A641
A637 88       DEY
A638 B1 5F    LDA ($5F),Y
A63A AA       TAX
A63B 88       DEY
A63C B1 5F    LDA ($5F),Y
A63E B0 D7    BCS $A617
A640 18       CLC
A641 60       RTS
```

Figura 1: routine n. 1

ta ad $A4A4 dove ricerca se il numero di linea esiste già.

Questo compito è svolto dalla routine locata a $A613 che legge in pagina zero il contenuto dei registri $14 e $15 che contengono il numero di linea corrente e li confronta con i numeri di linea già in memoria. Poi ritorna alla routine principale e salta alla locazione $A4ED dove risiede la routine per l'inserimento di una linea BASIC.

Se il numero di linea del programma originale dovesse esistere già, allora il contenuto di questa linea viene cancellato e quindi viene riscritto con il contenuto della linea del programma su disco.

Finito l'inserimento ritorna al BASIC e legge le locazioni $0302 e $0303 che contengono il puntatore di inizio programma (warm start). La routine appena descritta è visibile, disassemblata e commentata, nella prima figura, mentre la routine di MERGE è riportata in figura 2.

La prima istruzione si trova in $CF08 (#53000) ed è un salto alla subroutine locata a $E1D4; questa assume il nome del file ed i parametri di caricamento (file logico, device, indirizzo secondario).

Le istruzioni da $CF0B a $CF13 controllano se il file logico è diverso da uno per impedire il caricamento da registratore (LDA $BA e CMP #$01); nel caso sia uguale manda un messaggio di errore (JMP $A437).

Il salto a $AB1E permette di stampare il messaggio che inizia alla locazione di memoria contenuta nei registri A e Y (in questo caso $CFCB); il messaggio contiene la scritta MERGING. Il salto successivo a $F5C1 stampa di seguito al messaggio prima visto il nome del programma da caricare. In seguito vie-

ne controllato che non ci siano file aperti, per fare ciò viene eseguito il salto alla subroutine che inizia a $CFB3; nel caso siano aperti chiude tutti i file e ritorna a $CF25.

Memorizza in $B8 e $CFCA il numero del file logico corrente ed in $B9 l'indirizzo secondario, del nostro caso $00. Salto alla subroutine $FFC0 che apre un file logico a seconda dei parametri che gli sono stati passati.

Carica in X il contenuto $CFCA e salta a $FFC6 del KERNAL che apre il canale di input specificato nel registro X.

Adesso occorre modificare le locazioni di memoria $032C e $032D per evitare la chiusura del file logico; questo valore di solito contiene $F32F che chiude tutti i file aperti; per evitare questo si modifica il contenuto facendolo puntare a $CFB2 che contiene un RTS (ritorno da subroutine). Poi si modificano le locazioni che contengono il vettore di warm start per permettere alla routine

che punta a $A4A4, una volta tornata al BASIC, di sapere a quale indirizzo, deve andare per riprendere lo svolgimento del programma. Così le locazioni $0302 e $0303 conteranno il valore $CF8F.

In $CF8F c'è una istruzione JMP $CF51 che riporta il controllo del programma alla routine che eprmette l'inserimento delle linee lette dal file logico aperto. Le istruzioni da $CF51 a $CF8C leggono i caratteri da disco (JRS $FFCF), li memorizzano in $14 e $15 (numero riga) e nel buffer di input da $0200 (caratteri che formano la linea) per poi scriverli in memoria tramite l'istruzione JMP $A4A4 vista prima.

Sono di due tipi i controlli inseriti per vedere quando il programma è finito: il primo controlla se i due caratteri letti sono uguali a $00, il secondo se lo status ($90) è diverso a $00.

Se una delle condizioni è verificata salto a $CF92. Da $CF92 riportiamo a

```
1000 FFFF ;*****************************
1010 FFFF ;*                           *
1020 FFFF ;*     MERGE  PER COMMODORE  *
1030 FFFF ;*                           *
1040 FFFF ;*  I VALORI IN  ESADECIMALE *
1050 FFFF ;*                           *
1060 FFFF ;*     SONO RIFERITI AL C64  *
1070 FFFF ;*                           *
1080 FFFF ;*****************************
1090 FFFF ;*                           *
1100 FFFF ;* PREDISPONE  AL CARICAMENTO *
1110 FFFF ;*                           *
1120 FFFF ;*     DEL NUOVO  PROGRAMMA  *
1130 FFFF ;*                           *
1140 FFFF ;*****************************
1150 CF08 20 D4 E1      JSR SLPARA     ;PER C64 E VIC20 = E1D4
1160 CF0B A5 BA         LDA FA         ;PER C64 E VIC20 = BA
1170 CF0D C9 01         CMP #$01
1180 CF0F D0 05         BNE $CF16
1190 CF11 A2 09         LDX #$09
1200 CF13 4C 37 A4      JSR ERROR      ;PER C64 = A437,VIC20 = C437
1210 CF16 A9 CB         LDA #$CB
1220 CF18 A0 CF         LDY #$CF
1230 CF1A 20 1E AB      JSR STROUT     ;PER C64 = AB1E,VIC20 = CB1F
1240 CF1D 20 C1 F5      JSR $F5C1      ;STAMPA NOME FILE
1250 CF20 A9 0E         LDA #$0E
1260 CF22 20 B3 CF      JSR $CFB3
1270 CF25 85 8B         STA LA         ;PER C64 E VIC20 = B8
1280 CF27 8D CA CF      STA $CFCA
1290 CF2A A9 00         LDA #$00
1300 CF2C 85 B9         STA SA         ;PER C64 E VIC20 = B9
```

```
1310 CF2E 20 C0 FF      JSR $FFC0          ;APRE UN FILE LOGICO
1320 CF31 AE CA CF      LDX $CFCA
1330 CF34 20 C6 FF      JSR $FFC6          ;APERTURA CANALE INPUT
1340 CF37 A9 B2         LDA #$B2
1350 CF39 8D 2C 03      STA ICLALL(LB)     ;PER C64 E VIC20 = 032C
1360 CF3C A9 CF         LDA #$CF
1370 CF3E 8D 2D 03      STA ICLALL(HB)     ;PER C64 E VIC20 = 032D
1380 CF41 A9 8F         LDA #$8F
1390 CF43 8D 02 03      STA IMAIN(LB)      ;PER C64 E VIC20 = 0302
1400 CF46 A9 CF         LDA #$CF
1410 CF48 8D 03 03      STA IMAIN(HB)      ;PER C64 E VIC20 = 0303
1420 FFFF ;*****************************
1430 FFFF ;*                           *
1440 FFFF ;*      LETTURA DA DISCO E   *
1450 FFFF ;*                           *
1460 FFFF ;*  INSERIMENTO LINEA  BASIC *
1470 FFFF ;*                           *
1480 FFFF ;*****************************
1490 CF4B 20 CF FF      JSR $FFCF          ;PRENDE UN CARATTERE DAL DEVICE
1500 CF4E 20 CF FF      JSR $FFCF          ;PRENDE UN CARATTERE DAL DEVICE
1510 CF51 20 CF FF      JSR $FFCF          ;PRENDE UN CARATTERE DAL DEVICE
1520 CF54 85 14         STA LINNUM(LB)     ;PER C64 E VIC20 = 14
1530 CF56 20 CF FF      JSR $FFCF          ;PRENDE UN CARATTERE DAL DEVICE
1540 CF59 85 15         STA LINNUM(HB)     ;PER C64 E VIC20 = 15
1550 CF5B 05 14         ORA LINNUM(LB)
1560 CF5D F0 33         BEQ $CF92
1570 CF5F A5 90         LDA STATUS         ;PER C64 E VIC20 = 90
1580 CF61 D0 2F         BNE $CF92
1590 CF63 20 CF FF      JSR $FFCF          ;PRENDE UN CARATTERE DAL DEVICE
1600 CF66 85 14         STA LINNUM(LB)
1610 CF68 20 CF FF      JSR $FFCF          ;PRENDE UN CARATTERE DAL DEVICE
1620 CF6B 85 15         STA LINNUM(HB)
1630 CF6D A0 00         LDY #$00
1640 CF6F 20 CF FF      JSR $FFCF          ;PRENDE UN CARATTERE DAL DEVICE
1650 CF72 99 00 02      STA BUF,Y          ;PER C64 E VIC20 = 200
1660 CF75 A6 C5         LDX LSTX           ;PER C64 E VIC20 = C5
1670 CF77 E0 3F         CPX #$3F
1680 CF79 F0 17         BEQ $CF92
1690 CF7B C8            INY
1700 CF7C C9 00         CMP #$00
1710 CF7E D0 EF         BNE $CF6F
1720 CF80 98            TYA
1730 CF81 18            CLC
1740 CF82 69 04         ADC #$04
1750 CF84 85 0B         STA COUNT          ;PER C64 E VIC20 = 0B
1760 CF86 A5 90         LDA STATUS
1770 CF88 D0 08         BNE $CF92
1780 CF8A A4 0B         LDY COUNT
1790 CF8C 4C A4 A4      JMP INSLIN         ;PERC64 = A4A4,VIC20 = C4A4
1800 CF8F 4C 51 CF      JMP $CF51
1810 FFFF ;*****************************
1820 FFFF ;*                           *
```

```
1830 FFFF ;*  RESTORE VALORI A DEFAULT  *
1840 FFFF ;*                            *
1850 FFFF ;*   E CHIUSURA CANALI  I/O   *
1860 FFFF ;*                            *
1870 FFFF ;*****************************
1880 CF92 A9 83        LDA LBYTE        ;PER C64 =$83,VIC20 = #131
1890 CF94 8D 02 03     STA IMAIN(LB)
1900 CF97 A9 A4        LDA HBYTE        ;PER C64 =$A4,VIC20 =#196
1910 CF99 8D 03 03     STA IMAIN(HB)
1920 CF9C A9 2F        LDA LBYTE        ;PER C64 =$2F,VIC20 =#239
1930 CF9E 8D 2C 03     STA ICLALL(LB)
1940 CFA1 A9 F3        LDA HBYTE        ;PER C64 =$F3,VIC20 =#243
1950 CFA3 8D 2D 03     STA ICLALL(HB)
1960 CFA6 AD CA CF     LDA $CFCA
1970 CFA9 20 C3 FF     JSR $FFC3        ;CHIUDE IL FILE LOGICO
1980 CFAC 20 CC FF     JSR $FFCC        ;CHIUDE I CANALI I/O
1990 CFAF 4C 74 A4     JMP READY        ;PER C64 = A474,VIC20 = C474
2000 CFB2 60           RTS
2010 FFFF ;*****************************
2020 FFFF ;*                            *
2030 FFFF ;*    ROUTINE DI CONTROLLO    *
2040 FFFF ;*                            *
2050 FFFF ;*        FILE  APERTI        *
2060 FFFF ;*                            *
2070 FFFF ;*****************************
2080 CFB3 A6 98        LDX LPTND        ;PER C64 E VIC20 = 98
2090 CFB5 E0 00        CPX #$00
2100 CFB7 F0 10        BEQ $CFC9
2110 CFB9 DD 58 02     CMP BUF,X        ;PER C64 E VIC20 = 258
2120 CFBC D0 06        BNE $CFC4
2130 CFBE 38           SEC
2140 CFBF E9 01        SBC #$01
2150 CFC1 4C B3 CF     JMP $CFB3
2160 CFC4 CA           DEX
2170 CFC5 F0 02        BEQ $CFC9
2180 CFC7 D0 F0        BNE $CFB9
2190 CFC9 60           RTS
2200 CFCA 00           BRK
2210 FFFF ;*****************************
2220 FFFF ;*                            *
2230 FFFF ;*     STAMPA MESSAGGIO       *
2240 FFFF ;*                            *
2250 FFFF ;*****************************
2260 CFCB 4D 45 52     EOR $5245
2270 CFCE 47           BYT $47
2280 CFCF 49 4E        EOR #$4E
2290 CFD1 47           BYT $47
2300 CFD2 20 00 00     JSR $0000
2310 CFD5 00           BRK
2320
```

Figura 2: routine di merge commentata

valori di default le locazioni $0302, $0303 (warm start) e $032C, $032D (chiusura file logico). Salto alle subroutines del KERNAL $FFC3 che chiude il file logico, $FFCC che chiude i canali di input ed output. Infine salto a $A474 che stampa READY.

Il listato BASIC in figura 3 (per Commodore 64) contiene i DATA sotto forma esadecimale per facilitarne la digitazione ed evitare così possibili sbagli. La routine di conversione da esadecimale in decimale è da riga 128 a riga 130.

Per fare il merge tra due programmi caricare la routine proposta e dare il RUN; dopodiché caricare il programma più grosso come numero di blocchi e poi, per fare il MERGE con il programma su disco, digitare:

SYS (53000) "nome prg", 8

APPEND

Per capire il principio su cui si basa questa routine bisogna analizzare come è costruita una linea BASIC e cosa avviene prima e dopo il caricamento di un programma.

Una linea BASIC è composta da cinque parti fondamentali. La prima comprende due byte nei quali si trova il puntatore di inizio linea, poi ci sono altri due byte che contengono il numero di linea, il testo BASIC della linea, un byte che contiene uno zero che indica la fine della linea ed infine gli ultimi due byte che contengono il puntatore alla prossima linea. L'ultima linea conterrà negli ultimi due byte il valore zero per indicare che il programma è finito.

Quando si accende il computer o si da un comando di NEW tutti i valori contenuti nelle locazioni di memoria da $002B a $0032 contengono lo stesso valore e più precisamente $0801. Il significato di queste locazioni si può vedere in figura 4. All'atto del caricamento la routine del sistema operativo controlla il valore dell'indirizzo secondario; se l'indirizzo secondario non è presente il programma viene caricato a partire dall'inizio BASIC che corrisponde a $0801 (#2049) altrimenti se l'indirizzo

```
100 REM **************************
101 REM *                        *
102 REM *         MERGE          *
103 REM *                        *
104 REM **************************
105 REM *                        *
106 REM * AUTHOR SOFTWARE :      *
107 REM *                        *
108 REM *  GIANCARLO DE COBELLI  *
109 REM *                        *
110 REM **************************
111 REM *                        *
112 REM * VIC 20 + EXP.      SI  *
113 REM * COMMODORE 64       SI  *
114 REM * COMMODORE 4000     NO  *
115 REM * COMMODORE 8000     NO  *
116 REM * COMMODORE 16       NO  *
117 REM * COMMODORE PLUS 4   NO  *
118 REM *                        *
119 REM **************************

120 :
121 FOR K=53000 TO 53205:READ A$
122 :
123 REM **************************
124 REM * CONVERSIONE HEX TO DEC *
125 REM **************************
126 X=ASC(LEFT$(A$,1))-48:IF X>9 T
    HEN X=X-7
127 Y=ASC(RIGHT$(A$,1))-48:IF Y>9
    THEN Y=Y-7
128 A=X*16+Y:POKE K,A:NEXT
129 :
130 PRINT"[DOWN]DIGITARE SYS53000"
    CHR$(34)"NOME PRG"CHR$(34)",DE
    V."
131 PRINT"[DOWN]PER FARE IL MERGE.
    "
132 NEW :END
133 :
134 DATA 20,D4,E1,A5,BA,C9,01,D0
135 DATA 05,A2,09,4C,37,A4,A9,CB
136 DATA A0,CF,20,1E,AB,20,C1,F5
137 DATA A9,0E,20,B3,CF,85,B8,8D
138 DATA CA,CF,A9,00,85,B9,20,C0
139 DATA FF,AE,CA,CF,20,C6,FF,A9
140 DATA B2,8D,2C,03,A9,CF,8D,2D
141 DATA 03,A9,8F,8D,02,03,A9,CF
142 DATA 8D,03,03,20,CF,FF,20,CF
143 DATA FF,20,CF,FF,85,14,20,CF
144 DATA FF,85,15,05,14,F0,33,A5
145 DATA 90,D0,2F,20,CF,FF,85,14
146 DATA 20,CF,FF,85,15,A0,00,20
147 DATA CF,FF,99,00,02,A6,C5,E0
148 DATA 3F,F0,17,C8,C9,00,D0,EF
149 DATA 98,18,69,04,85,0B,A5,90
150 DATA D0,08,A4,0B,4C,A4,A4,4C
151 DATA 51,CF,A9,83,8D,02,03,A9
152 DATA A4,8D,03,03,A9,2F,8D,2C
153 DATA 03,A9,F3,8D,2D,03,AD,CA
154 DATA CF,20,C3,FF,20,CC,FF,4C
155 DATA 74,A4,60,A6,98,E0,00,F0
156 DATA 10,DD,58,02,D0,06,38,E9
157 DATA 01,4C,B3,CF,CA,F0,02,D0
158 DATA F0,50,00,4D,45,52,47,49
159 DATA 4E,47,20,80,00,00
```

Figura 3: i DATA in forma esadecimale a programma BASIC strutturato

secondario corrisponde ad uno viene letto nei primi due byte del programma su disco l'indirizzo di caricamento, poiché tutte le volte che si salva un programma, i primi due byte sono riservati all'indirizzo di partenza del programma (ecco perché nei programmi scritti in linguaggio macchina viene richiesto l'indirizzo secondario).

Finito il caricamento, nelle locazioni $002D e $002E si avrà l'indirizzo di fine testo BASIC o inizio variabili poiché le variabili sono in coda al programma caricato.

Ora, utilizzando tutte queste informazioni è facile dedurre come realizare un APPEND, cioè aggiungere in coda al programma memorizzato un programma letto da disco. Riferendoci alla figura 5 dove c'è il disassemblato, le prime due istruzioni servono per impostare il computer in caricamento (la locazione $0A deve contenere $00 per il LOAD o $01 per il VERIFY); salto alla routine già analizzata prima (JSR $E1D4) che predispone i parametri di collegamento con la periferica.

Le istruzioni da $CF11 a $CF14 servono per memorizzare nei registri A, X, Y i parametri richiesti dalla subroutine del KERNAL $FFD5 già analizzata nello scorso numero (carica un programma da periferica). In A mettiamo il valore $00 per il LOAD ($01 per il VERIFY) con l'istruzione LDA $0A. In X e Y bisogna memorizzare i valori da dove deve iniziare a memorizzare il programma che viene letto da disco; poiché bisogna attaccarlo in coda in X e Y verrà memorizzato il contenuto delle locazioni $002D e $002E. In $002D c'è il byte basso della locazione di partenza del nuovo programma e quindi dovremo decrementarlo di due (SBC #$02) perché abbiamo visto che il programma in memoria alla fine viene chiuso con due zeri che si dovranno togliere per inserire il nuovo puntatore alla prossima linea del programma che verrà aggiunto.

La subroutine $A533 serve per linkare (legare) tutto il programma che viene aggiunto (aggiornamento dei puntatori alle righe successive). Da $CF22 e $CF39, il programma testa tutti i puntatori che rimandano alla nuova linea per trovare la locazione di fine programma. Utilizza per questo scopo le locazioni $57 e $58 che sono a disposizione per la memorizzazione di dati transienti; il confronto avviene con il numero zero per i motivi analizzati prima.

Se il puntatore non contiene zeri, salta alla routine $CF51-$CF60 che memorizza in $57 e $58 il prossimo puntatore e ritorna a $CF30 a fare i confronti. Se il primo byte del puntatore (byte basso) è uguale a zero passa ad un ulteriore confronto sul secondo byte (byte alto) per vedere che valore contiene; se è diverso da zero salto alla subroutine di prima per lo stesso motivo. Altrimenti memorizza nelle locazioni da $002D a $0032 la locazione di fine programma ($CF3D-$CF50); l'istruzione RTS ritorna il controllo al BASIC.

In figura 6 troviamo il programma BASIC strutturato con lo stesso principio del programma in figura 3. Per l'utilizzo del programma digitare:

SYS (53000) "nome prg", 8 e dare il RUN.

Dopo aver caricato il primo programma in modo normale caricare il secondo utilizzando la SYS. Per chi possiede un assemblatore è molto più comodo digitare queste routines direttamente in codice mnemonico copiando i disassemblati. Questo elimina il problema di caricare per prima la routine e dopoi programmi poiché essendo locate a $CF08 non influiscono assolutamente con i programmi in memoria.

A secondo del tipo di computer su cui si vuole digitare il programma si deve inserire l'adatto valore numerico facendo riferimeto ai commenti a fianco di ogni label. Se il programma deve essere digitato in BASIC occorrerà cambiare i DATA in maniera opportuna e fare attenzione anche al fatto che i due programmi per il Commodore 64 partono dalla locazione #53000 e quindi occorrerà anche provvedere a rilocare più in basso le routine non dimenticandosi di modificare anche i JMP e JSR.

LABEL	HEX	DEC	DESCRIZIONE
TXTTAB	2B-2C	43-44	INIZIO BASIC
VARTAB	2D-2E	45-46	INIZIO VARIAB.
ARYTAB	2F-30	47-48	INIZIO MATRICI
STREND	31-32	49-50	FINE MATRICI

Figura 4: locazioni e loro significato

```
10 FFFF     ;********************************
11 FFFF     ;*                              *
12 FFFF     ;*     APPEND PER COMMODORE     *
13 FFFF     ;*                              *
14 FFFF     ;*  I VALORI IN  ESADECIMALE    *
15 FFFF     ;*                              *
16 FFFF     ;*     SONO RIFERITI AL C64     *
```

```
17 FFFF   ;*                              *
18 FFFF   ;********************************
19 FFFF   ;*                              *
20 FFFF   ;* PREDISPONE  AL  CARICAMENTO  *
21 FFFF   ;*                              *
22 FFFF   ;* IN CODA  AL PRO IN MEMORIA   *
23 FFFF   ;*                              *
24 FFFF   ;********************************
25 CF08   A9 00       LDA #$00
26 CF0A   85 0A       STA VERCK        ;PER 64 E VIC20 = 0A
27 CF0C   20 D4 E1    JSR SLPARA       ;PER 64 = E1D4,VIC20 = E1D1
28 CF0F   A5 2D       LDA VARTAB(LB)   ;PER 64 E VIC20 = 2D
29 CF11   38          SEC
30 CF12   E9 02       SBC #$02
31 CF14   AA          TAX
32 CF15   A5 2E       LDA VARTAB(HB)   ;PER 64 E VIC20 = 2E
33 CF17   E9 00       SBC #$00
34 CF19   A8          TAY
35 CF1A   A5 0A       LDA VERCK
36 CF1C   20 D5 FF    JSR $FFD5        ;RICHIAMO ROUTINE DI LOAD
37 CF1F   20 33 A5    JSR LNKPRG       ;PER 64 = A533,VIC20 = C533
38 FFFF   ;********************************
39 FFFF   ;*                              *
40 FFFF   ;*    RICERCA FINE PROGRAMMA    *
41 FFFF   ;*                              *
42 FFFF   ;********************************
43 CF22   A5 2D       LDA VARTAB(LB)
44 CF24   A4 2E       LDY VARTAB(HB)
45 CF26   38          SEC
46 CF27   E9 02       SBC #$02
47 CF29   85 57       STA TEMPF1(1)    ;PER 64 E VIC20 = 57
48 CF2B   98          TYA
49 CF2C   E9 00       SBC #$00
50 CF2E   85 58       STA TEMPF1(2)    ;PER 64 E VIC20 = 58
51 CF30   A0 00       LDY #$00
52 CF32   B1 57       LDA(TEMPF1(1)),Y
53 CF34   D0 1B       BNE $CF51
54 CF36   C8          INY
55 CF37   B1 57       LDA(TEMPF1(1)),Y
56 CF39   D0 1B       BNE $CF51
57 FFFF   ;********************************
58 FFFF   ;*                              *
59 FFFF   ;*AGGIORNAMENTO  DEI PUNTATORI*
60 FFFF   ;*                              *
61 FFFF   ;********************************
62 CF3B   A5 57       LDA TEMPF1(1)
63 CF3D   18          CLC
64 CF3E   69 02       ADC #$02
65 CF40   85 2D       STA VARTAB(LB)
66 CF42   85 2F       STA ARYTAB(LB)   ;PER 64 E VIC20 = 2F
67 CF44   85 31       STA STREND(LB)   ;PER 64 E VIC20 = 31
68 CF46   A5 58       LDA TEMPF1(2)
```

```
69 CF48   69 00       ADC #$00
70 CF4A   85 2E       STA VARTAB(HB)
71 CF4C   85 30       STA ARITAB(HB)   ;PER 64 E VIC20 = 30
72 CF4E   85 32       STA STREND(HB)   ;PER 64 E VIC20 = 32
73 CF50   60          RTS
74 FFFF   ;*****************************
75 FFFF   ;*  AGGIORNAMENTO  PUNTATORE  *
76 FFFF   ;*                            *
77 FFFF   ;*    PROSSIMA LINEA BASIC    *
78 FFFF   ;*****************************
79 CF37   B1 57       LDA(TEMPF1(1)),Y
80 CF55   85 59       STA TEMPF1(3)    ;PER 64 E VIC20 = 59
81 CF57   C8          INY
82 CF58   B1 57       LDA(TEMPF1(1)),Y
83 CF5A   85 58       STA TEMPF1(2)
84 CF5C   A5 59       LDA TEMPF1(3)
85 CF5E   85 57       STA TEMPF1(1)
86 CF60   4C 30 CF    JMP $CF30
87 CF63   00          BRK
```

Figura 5: disassemblato

```
100 REM **************************
101 REM *                        *
102 REM *         APPEND         *
103 REM *                        *
104 REM **************************
105 REM *                        *
106 REM * AUTHOR SOFTWARE :      *
107 REM *                        *
108 REM *  GIANCARLO DE COBELLI  *
109 REM *                        *
110 REM **************************
111 REM *                        *
112 REM * VIC 20 +EXP.       SI  *
113 REM * COMMODORE 64       SI  *
114 REM * COMMODORE 4000     NO  *
115 REM * COMMODORE 8000     NO  *
116 REM * COMMODORE 16       NO  *
117 REM * COMMODORE PLUS 4   NO  *
118 REM *                        *
119 REM **************************
120 :
121 FOR K=53000 TO 53091:READ A$
122 :
123 REM **************************
124 REM * CONVERSIONE HEX TO DEC *
125 REM **************************
126 X=ASC(LEFT$(A$,1))-48:IF X>9 T
    HEN X=X-7
127 Y=ASC(RIGHT$(A$,1))-48:IF Y>9
    THEN Y=Y-7
128 A=X*16+Y:POKE K,A:NEXT
129 :
130 PRINT"[DOWN]DIGITARE SYS53000"
    CHR$(34)"NOME PRG"CHR$(34)",DE
    V."
131 PRINT"[DOWN]PER FARE L'APPEND.
    "
132 NEW :END
133 :
134 DATA A9,00,85,0A,20,D4,E1,A5
135 DATA 2D,38,E9,02,AA,A5,2E,E9
136 DATA 00,A8,A5,0A,20,D5,FF,20
137 DATA 33,A5,A5,D2,A4,2E,38,E9
138 DATA 02,85,57,98,E9,00,85,58
139 DATA A0,00,B1,57,D0,1B,C8,B1
140 DATA 57,D0,16,A5,57,18,69,02
141 DATA 85,2D,85,2F,85,31,A5,58
142 DATA 69,00,85,2E,85,30,85,32
143 DATA 60,A0,00,B1,57,85,59,C8
144 DATA B1,57,85,58,A5,59,85,57
145 DATA 4C,30,CF,00
```

Figura 6: i data in forma esadecimale e programma BASIC strutturato

UN FENOMENO FISICO

Il comportamento Elastico dei Corpi

di Mauro Massetti

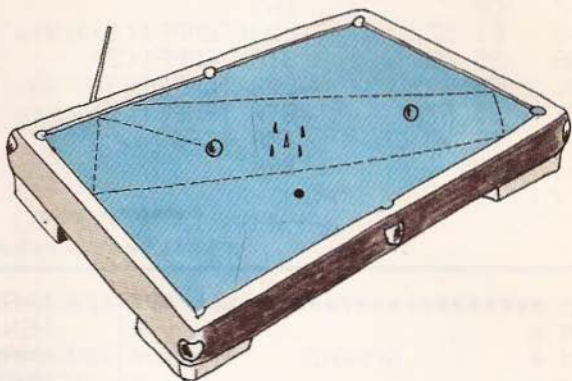
Come è ormai ampiamente risaputo, sebbene non tutti accettino questo dato di fatto, il Computer si dimostra di grandissima utilità nella didattica in generale ed in modo particolare nell'aiuto alla comprensione di alcuni fenomeni fisici o leggi matematiche, proprio perché la velocità di elaborazione e di simulazione ci permette di "toccare con mano" i vari fenomeni.

Queste due materie sono sempre state classificate come "fredde" anche perché sono, nella quasi totalità, molto difficili da illustrare a parole, da capire concettualmente e da valutare nelle loro più varie sfumature. Prendiamo ad esempio un fenomeno fisico che, ad insaputa di molti, è il vero protagonista delle domeniche:

Il fenomeno elastico

È infatti grazie alle proprietà elastiche dei corpi che il pallone passa da un piede ad un altro piuttosto che ad un altro ancora, rimbalza sul campo di calcio, vola per un colpo di testa, batte sulla traversa e si insacca in rete.

È comunque molto difficile rendersi conto di come varia il presentarsi di un fenomeno al variare di alcuni parametri, avendo sott'occhio solamente le formule che lo traducono matematicamente, e lungo e laborioso eseguire manualmente i calcoli necessari per un numero di esempi significativi per la comprensione dello stesso. La mancata comprensione porta spesso ad errori concettuali come, nel caso specifico, quello di credere che sia più elastico un urto fra due biglie di gomma che fra due di acciaio.

A questo proposito si vuole spiegare che l'idea principale di "urto" è che il moto delle particelle collidenti, o di almeno una di esse, varii piuttosto rapidamente e che sia possibile una separazione piuttosto netta di tempi "prima dell'urto" e "dopo l'urto".

Quando una mazza, per esempio, colpisce una palla da baseball, l'inizio e la fine dell'urto possono essere determinati con sufficiente precisione: la mazza resta in contatto con la palla per un intervallo di tempo molto breve in confronto al tempo per il quale si osserva la palla e durante l'urto la mazza esercita una grande forza sulla palla. Questa forza varia nel tempo in un modo così complesso che la si può misurare con difficoltà. Durante l'urto tanto la palla quanto la mazza si deformano.

Le forze agenti per un tempo breve in confronto al tempo di osservazione del sistema sono chiamate "impulsive". Gli urti sono usualmente classificati a seconda che in essi si conservi o meno l'energia cinetica (è la forma di energia proria del moto la cui espressione fisica è $T = 1/2^{*} mv^2$, dove "m" è la massa del corpo e "v" è la sua velocità). Quando questa si conserva, l'urto viene detto ELASTICO; altrimenti è ANELASTICO.

Gli urti (o meglio le collisioni) fra particelle atomiche, nucleari sono talvolta elastiche: questi infatti sono i soli urti veramente elastici che si conoscono. Gli urti fra corpi estesi sono in un certo senso sempre anelastici; spesso però si possono trattare, senza introdurre grossi errori, come approssimativamente elastici, come ad esempio quelli che avvengono fra sfere di avorio o di vetro.

Infatti accade che, quando due corpi si urtano fra loro, parte dell'energia che possiedono venga, per così dire, "usata" nella deformazione degli stessi. In conseguenza a ciò appare chiaro che due corpi, definiti comunemente "rigidi", diano origine ad un fenomeno elastico in senso stretto.

Per capirci meglio si consideri, su un tavolo da biliardo, dapprima due sfere di gomma; potendo riprendere l'esperimento onde rivederlo sufficientemente rallentato si potrebbe notare che, urtandosi, ciascuna di esse, per un istante si deformerà schiacciandosi sull'area di contatto ed alla forza con la quale si respingeranno dovrà essere sottratta la parte di energia spesa per la deformazione; se ora si considerano due palle da biliardo vere e proprie e si ripete quanto sopra, si nota che la deformazione da esse subita risulta minima e, quindi, minima è la parte di energia che viene sottratta alla forza con la quale esse si respingono.

Come esempio si passa a calcolare il moto di una sfera che, rimbalzando

sotto l'azione della forza di gravità, si sposta allo stesso tempo in direzione orizzontale. Per semplicità, che per altro non altera le relazioni fisiche del fenomeno in atto, si suppone che non vi siano attriti, ne dovuti al mezzo nel quale si muove la sfera ne al contatto fra le sfere, e che il coefficiente di rimbalzo sia dato da quella parte di energia che non viene alterata dall'impatto.

Si presupporrà nel programma noto lo spostamento verticale iniziale, vale a dire l'altezza originale sul terreno, nonché la velocità orizzontale V e il numero di volte N che la palla rimbalza. Sarà anche noto il coefficiente di rimbalzo C, rapporto tra la velocità verticale subito dopo l'urto e velocità subito prima.

Il calcolo

Per calcolare la posizione in tempi diversi, si sceglierà prima un piccolo incremento di tempo D e si userà poi la seguente legge, che in fisica vale per ogni incremento di tempo:

T(I + 1) = T(I) + D
X(I + 1) = X(I) + V*D
Z(I + 1) = Z(I)-G*D
Y(I + 1) = Y(I) + .5*(Z(I) + Z(I + 1))*D

dove X si riferisce allo spostamento orizzontale, zero in origine all'inizio del fenomeno, Z è la velocità verticale, che è anch'essa zero all'inizio del problema, Y l'altezza sul terreno e G l'accelerazione di gravità (9,81 m/s²). Gli indici I e I + 1 indicano i valori delle diverse variabili rispettivamente all'inizio e alla fine dell'incremento di tempo.

Se durante questo incremento di tempo si verifica un rimbalzo, le formule di calcolo vanno un po' modificate; una condizione di rimbalzo è caratterizzata da un valore negativo di Y (I + 1), il che è fisicamente impossibile. Quando si verifica detta condizione, si ricalcola Z(I + 1) e Y(I + 1), nel modo che segue. Prima si trova il tempo necessario alla palla per toccare il suolo, partendo dalla posizione all'inizio del tempo I-mo.

Chiamando questo tempo D1 si avrà, con una proporzione lineare:

D1/D = Y(I)/Y(I)-Y(I + 1)

e scrivendo in BASIC:

D1 = D*Y(I)/(Y(I)-Y(I + 1))

Si può ora calcolare la velocità verticale immediatamente prima dell'urto e sarà:

Z = Z(I)-G*D1

così che la velocità verticale subito dopo l'urto sarà:

Z1 = -C*(Z(I)-G*D1)

e la velocità vericale alla fine dell'incremento di tempo:

Z(I + 1) = Z1-G*(D-D1)

e lo spostamento in senso verticale alla fine dell'incremento di tempo sarà:

Y(I + 1) = .5*(Z1 + Z(I + 1))*(D-D1)

Schema del programma

Ora si hanno abbastanza informazioni a disposizione per tracciare lo schema del programma completo:

1- leggere H,V,N,C,D

- se H = O il calcolo termina
- se ad H è assegnato un valore positivo, andare al passo successivo

2- inizializzare tutti i parametri

I = 1 (I è il contatore d'incremento)
X(I) = O
B = O (B è il contatore dei rimbalzi)
Z(I) = O
T(I) = O Y(I) = H

3- calcolare spostamento orizzontale e verticale e velocità verticale per ogni incremento di tempo, usando le formule esaminate precedentemente

4- se la palla urta il terreno durante l'incremento di tempo, controllare se si tratta di una condizione di rimblazo o se invece termina il programma:

- condizione di rimbalzo (B < N) bisogna ricalcolare sia la velocità che lo spostamento verticale per tener conto del rimbalzo; bisogna poi incrementare il relativo contatore (B = B + 1), e si passa quindi al successivo incremento di tempo.
- condizione terminale (B = N) si ricava l'ultimo tempo, e lo spostamento orizzontale quando la palla urta il terreno

5- stampare i valori finali di X e T, seguiti da una tabella completa di T,X,Y,Z

6- diagrammare Y in funzione di T

7- tornare al punto 1.

Il programma BASIC

Qui di seguito è riportato il listato completo del programma BASIC per la "simulazione del rimbalzo di una sfera" ed un commento sui blocchi più importanti nonché; la flow-chart dello stesso e la cross-reference.

REMarcks

24- dimensionamento dei vettori

31-130 input dei dati: per alcuni di essi è stato fissato un massimo per ragioni grafiche; per quanto riguarda il numero dei rimbalzi e l'incremento di tempo, è bene tener presente che data la rappresentazione grafica degli stessi su un'unica videata e l'approssimazione della rappresentazione, bisogna mantenere valori rispettivamente non superiori a 4 per i rimbalzi e inferiori a 0.1 per l'incremento di tempo. Per quest'ultimo bisogna inoltre tener presente che le formule usate hanno valore solo per incrementi sufficientemente piccoli il che porta il "range" disponibile ad essere compreso fra 1 e 0.1.

180-190 inizializzazione dei parametri

240-420 calcola i parametri per ogni incremento di tempo, controlla se vi è un rimbalzo e, nel caso l'evento sia accaduto, inverte la direzione di moto e calcola i parametri relativi all'ultimo contatto. In questo blocco viene utilizzata la subroutine allocata dalla linea **760** alla linea **790**

450-510 visualizzazione alfanumerica dei dati impostati e, a richiesta, calcolati

560-590 routine di grafica. Visualizza graficamente i dati precedentemente calcolati. In questo blocco si usano le subroutines allocate rispettivamente dalle linee **1410, 1440, 10020** alle linee **1430, 1460, 10150**

599-610 torna all'inizio e termina l'esecuzione

760-790 subroutine di calcolo dei parametri: tempo, coordinate e velocità

1410-1430 posiziona la sfera sullo schermo; questa subroutine utilizza il primo dei 17 caratteri speciali ridefiniti

1440-1460 segna la posizione precedentemente occupata dalla sfera; questa routine utilizza i rimanenti 16 caratteri ridefiniti scegliendone uno a mezzo della subroutine allocata dalla linea **20010** alla linea **20090**

10020-10150 subroutine di ricopiatura caratteri dalla ROM alla RAM e di creazione dei 17 caratteri speciali ridefiniti. I dati relativi ai caratteri utente sono collocati dalla linea **10160** alla linea **10320**

20010-20090 sceglie quale dei 16 caratteri speciali è più adatto a rappresentare la posizione precedentemente assunta dalla sfera. Grazie a questa metodologia si riesce ad aumentare considerevolmente la precisione grafica del programma senza dover far uso della grafica ad alta risoluzione

21000-21100 intestazione.

L'utilizzo dei caratteri predefiniti, rispetto alla grafica ad alta risoluzione, ha come vantaggio una maggiore velocità di visualizzazione grafica anche se impone un'attesa, un po' noiosa, nella fase di ricopiatura da ROM a RAM.

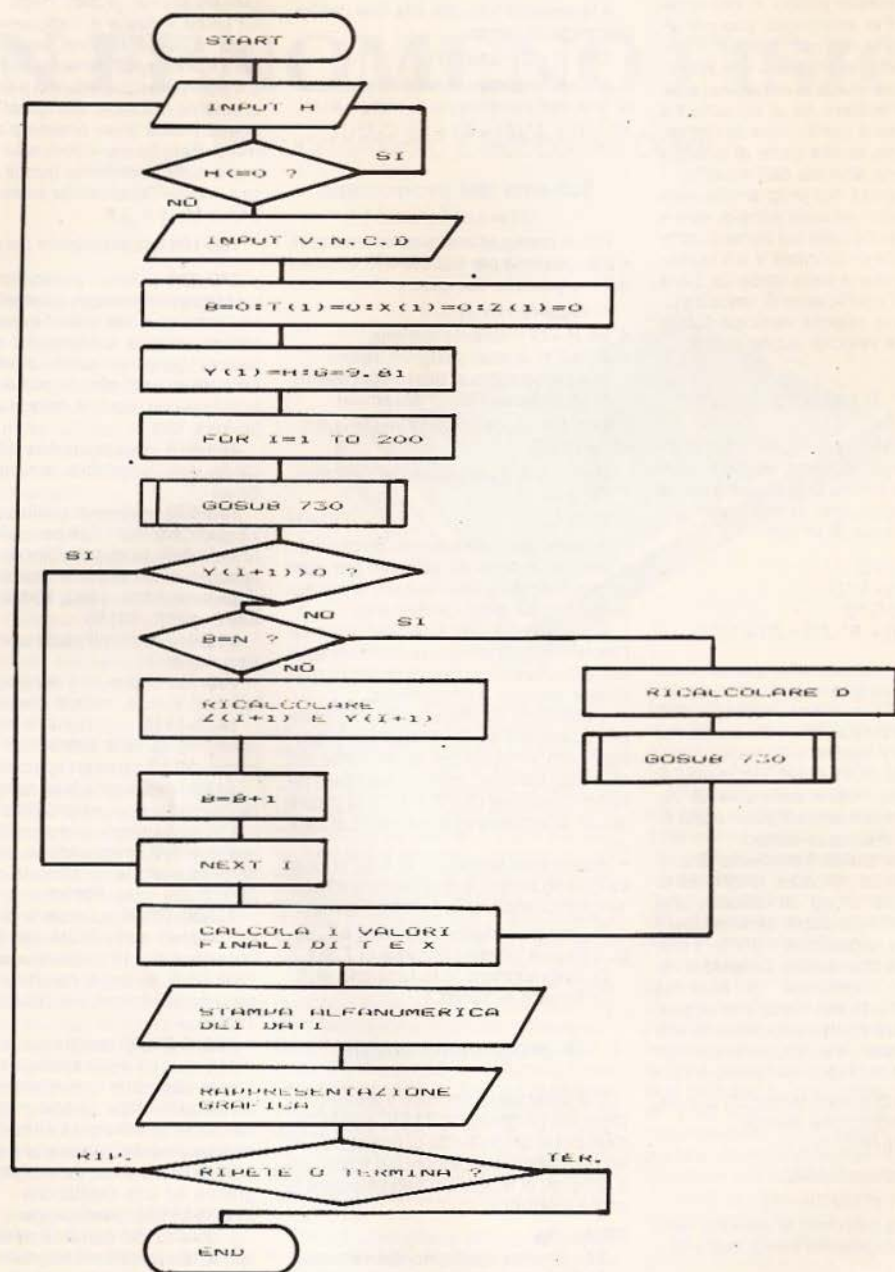



Figura 1: flow chart del blocco principale del programma.

10020

COPIATURA ROM> RAM
E CREAZIONE CARATTERI
SPECIALI UTENTE

RETURN

730

CALCOLARE
T(I+1)=X(I+1)
Z(I+1)=Y(I+1)

RETURN

20000

SCELTA CARATTERE
GRAFICO

RETURN

1410

POSIZIONA LA SFERA

RETURN

21000

INTESTAZIONE

RETURN

1440

POSIZIONE PRECEDENTE

RETURN

Figura 2: rappresentazioni in flow chart delle sei subroutine richiamate durante l'esecuzione del programma

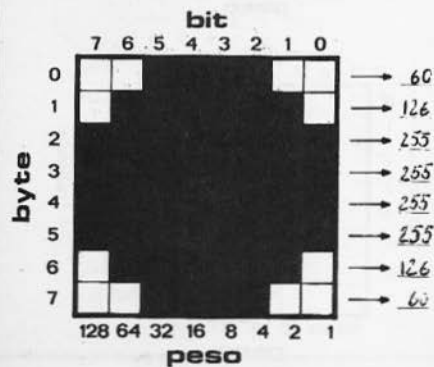
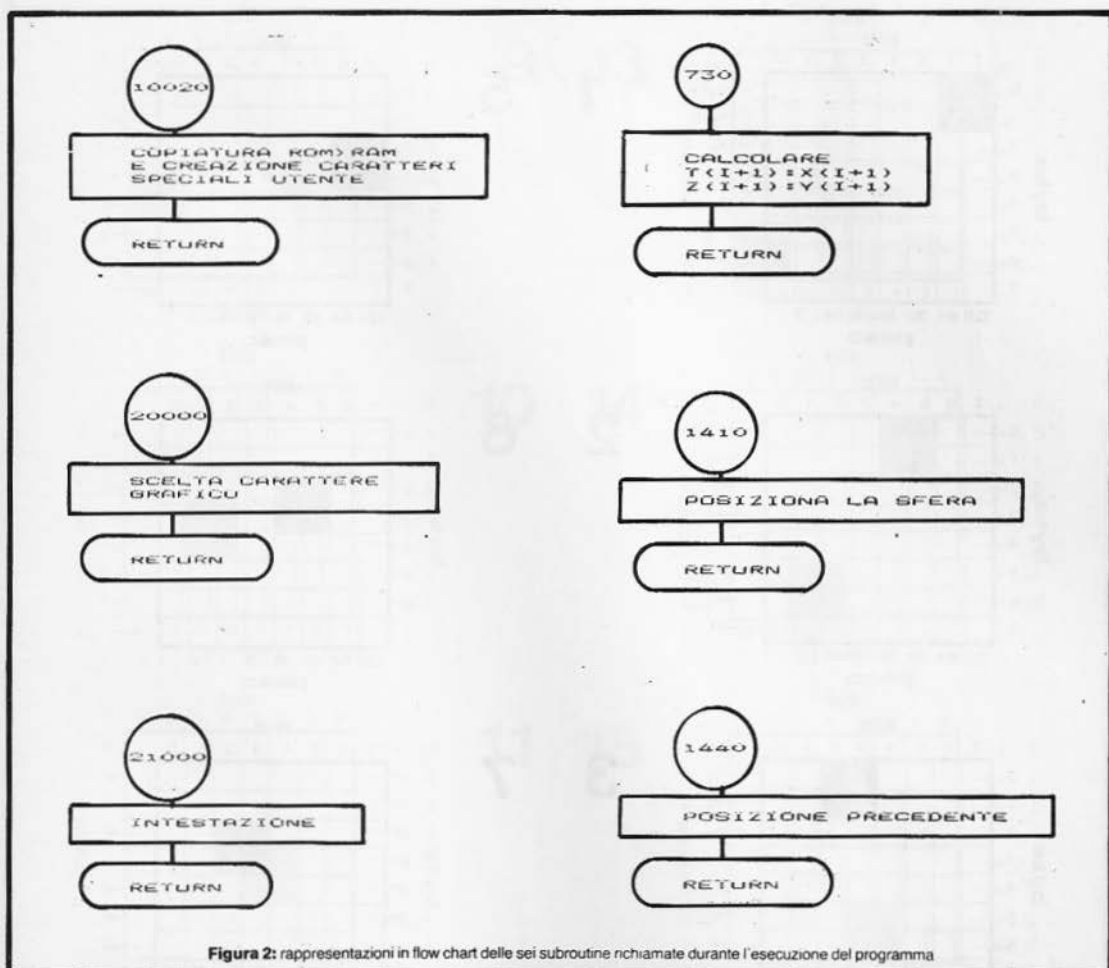
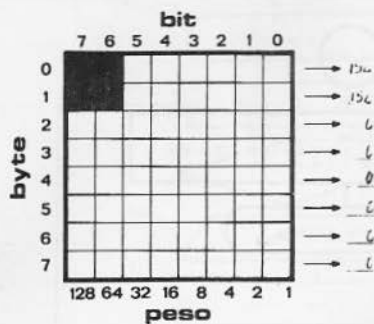
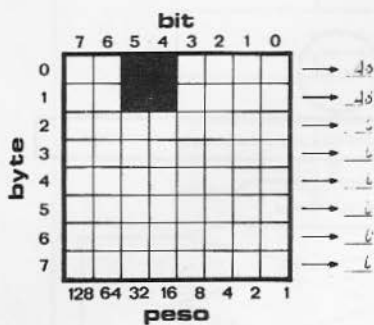
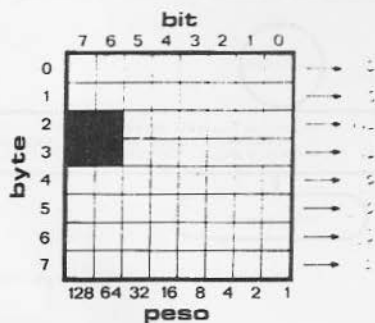
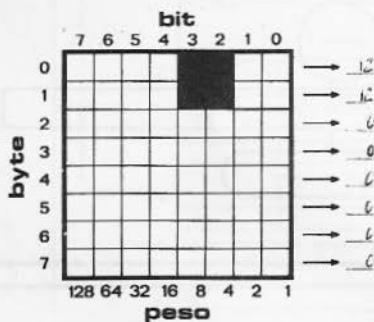
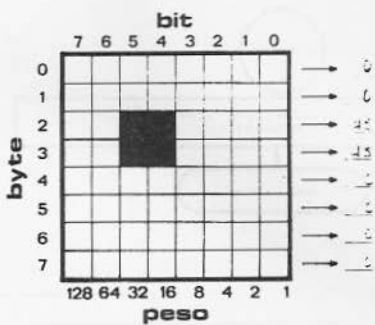
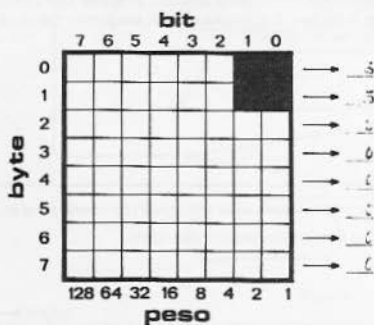
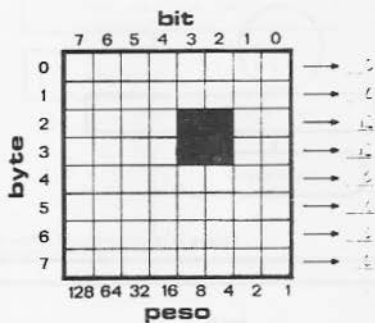
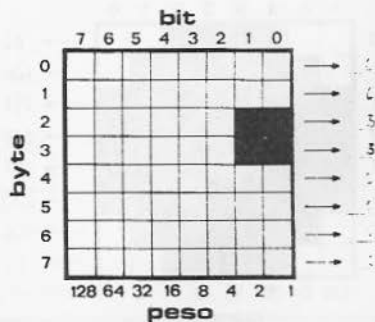
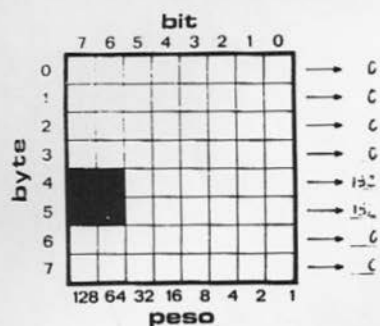
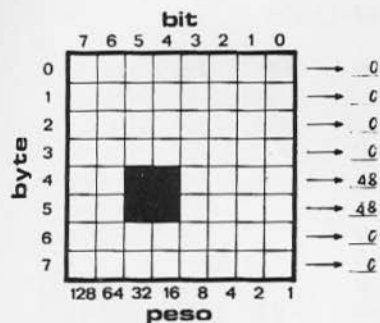
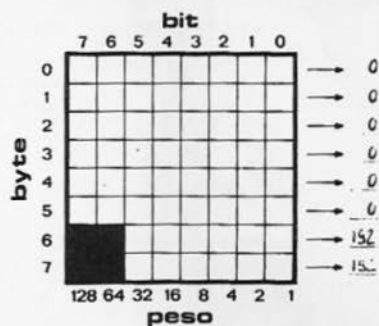
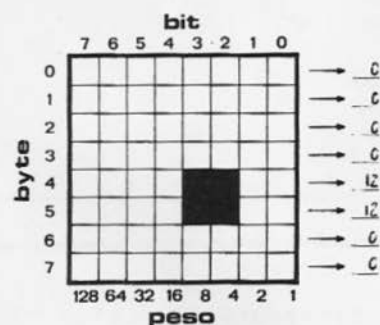
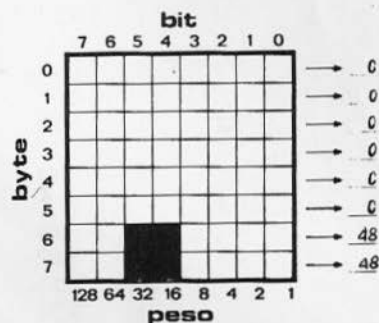
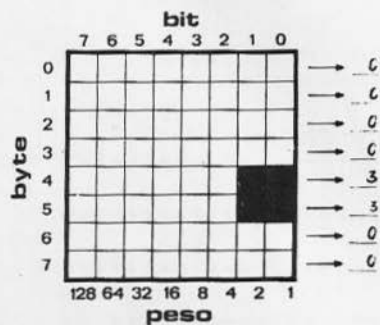
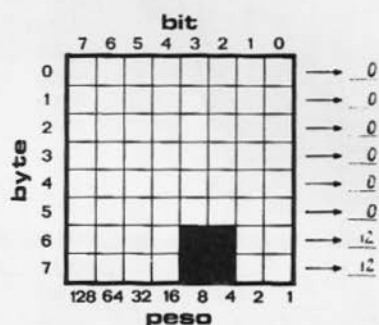
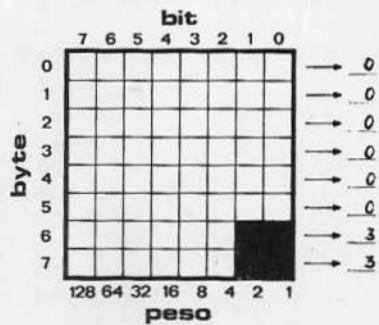
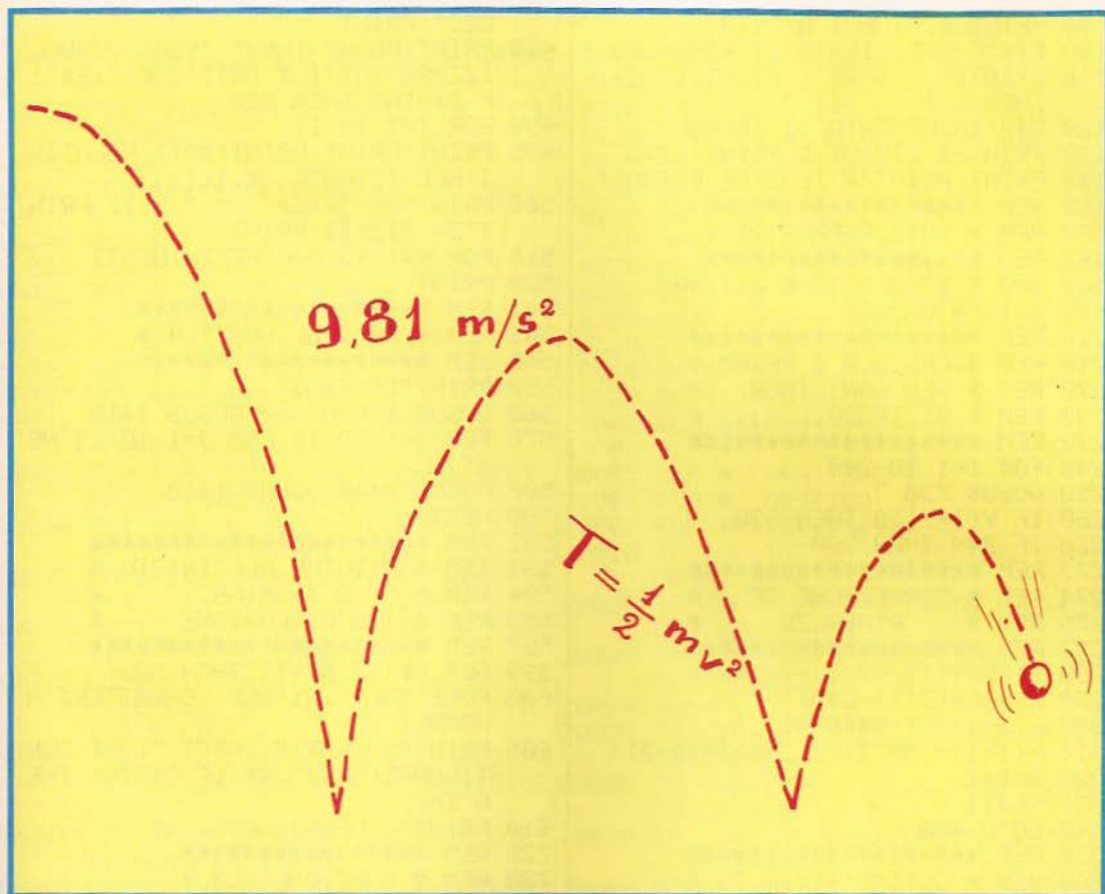



Figura 3: ecco come si costruisce un carattere speciale. In questa figura viene creata la sfera che faremo rimbalzare. Ecco anche come si eseguono i calcoli per ottenere i valori decimali da inserire nei DATA.

Figura 4: con la medesima logica di costruzione della figura precedente si creano i codici punti che servono per tracciare la traiettoria della sfera.

segue →

1
2
3
4
5
6
7
8
bit
7 6 5 4 3 2 1 0
byte
0 1 2 3 4 5 6 7
128 64 32 16 8 4 2 1
peso

9
13
10
14
11
15
12
16
bit
7 6 5 4 3 2 1 0
byte
0 1 2 3 4 5 6 7
128 64 32 16 8 4 2 1
peso

```
PROGRAMMA : FENOMENO EL.

-------*---------------- - - -
VAR.   ! LINEA DEL PROGRAMMA
-------*---------------- - - -
       !
A$     !    31    40   450   599
B      !   180   270   320
B$     !    60    70   451
BY     ! 10080 10120 10140
C      !   110   290   453
C$     !    80    90   452
CH     !  1450 10060 10120 10140 20090
CP     !   420  1410  1440
D      !   130   280   300   310   380   760
       !   770   780   790
D$     !   100   110   453
D1     !   280   290   300   310
DI     !   130   454
E$     !   120   130   454
G      !   190   290   300   780
H      !    40    50   190   450
I      !   240   260   280   290   300   310
       !   330   380   400   490   495   500
       !   510   560   570   590   760   770
       !   780   790  1410  1440 10035 10037
       ! 10038 21100
I1     !   400   410   420   490   570
J      !   570 10036 10037 10038
K      !   510
N      !    90   270   452
NU     ! 10100 10120
PO     !  1410  1420  1445  1450
PX     ! 20010 20020 20030 20040 20090
PY     ! 20050 20060 20070 20080 20090
S$     !   480   605
T(     !    24   180   410   495   760
T1     !   410   470
V      !    70   451   770
X(     !    24   180   420   500   770  1410
       !  1440
X1     !   420   460
XP     !  1440  1445 20010 20020 20030
Y(     !    24   190   260   280   310   380
       !   500   790  1410  1440
YP     !  1440  1445 20050 20060 20070
Z(     !    24   180   290   300   310   500
       !   780   790
Z1     !   290   300   310
```

```
10 REM *********************
11 REM *   SIMULAZIONE       *
12 REM *  DEL RIMBALZARE     *
13 REM *   DI UNA SFERA      *
14 REM *********************
15 REM * AUTHOR:             *
16 REM *    MAURO MASSETTI   *
17 REM *********************
18 REM * SOLO PER C-64       *
19 REM *********************
20 REM
21 REM *********************
22 REM *  DIMENS.VETTORI     *
23 REM *********************
24 PRINT"[CLEAR]":DIM  X(202),Y(2
   02),Z(202),T(202):GOSUB 21000
25 REM ********************
27 REM * INPUTAZIONE DATI *
29 REM ********************
30 REM
31 A$="ALTEZZA INIZIALE DELLA SFE
   RA "
40 PRINTA$;:INPUT H:PRINT:IF H>=2
   5 THEN H=24.99999
50 IF H<=0 THEN 40
60 B$="VELOCITA' ORIZZ. INIZIALE
   "
70 PRINTB$;:INPUT V:PRINT:IF V>10
    THEN V=10
80 C$="NUMERO DEI RIMBALZI "
```

```
 90 PRINTC$;:INPUT N:PRINT
100 D$="COEFFICIENTE DI RIMBALZO "
110 PRINTD$;:INPUT C:PRINT:IF C>1
    THEN C=1
120 E$="INCREMENTO DI TEMPO "
130 PRINTE$;:INPUT D:PRINT:DI=D
140 PRINT:PRINT"ATTENDERE PREGO!"
158 REM *****************
160 REM * INIZ.PARAMETRI *
162 REM *****************
180 B=0:T(1)=0:X(1)=0:Z(1)=0
190 Y(1)=H:G=9.81
218 REM ******************
219 REM * CALCOLA I PARAM *
220 REM * PER OGNI INCR.  *
230 REM * DI TEMPO        *
232 REM ******************
240 FOR I=1 TO 200
250 GOSUB 730
260 IF Y(I+1)>0 THEN 330
270 IF B=N THEN 360
273 REM ******************
274 REM * CORREZIONE DEL  *
275 REM *    RIMBALZO     *
277 REM ******************
280 D1=D*Y(I)/(Y(I)-Y(I+1))
290 Z1=-C*(Z(I)-G*D1)
300 Z(I+1)=Z1-G*(D-D1)
310 Y(I+1)=.5*(Z1+Z(I+1))*(D-D1)
320 B=B+1
330 NEXTI
340 GOTO 400
358 REM ******************
360 REM * ULTIMO RIMBALZO *
362 REM ******************
380 D=D*Y(I)/(Y(I)-Y(I+1))
390 GOSUB 730
400 I1=I
410 T1=T(I1)
420 X1=X(I1):CP=39/X1
438 REM **********************
440 REM *  OUTPUT NUMERICO    *
442 REM **********************
450 PRINT"[CLEAR]":PRINT:PRINT:PRI
    NTA$;H:PRINT
451 PRINTB$;V:PRINT
452 PRINTC$;N:PRINT
453 PRINTD$;C:PRINT
454 PRINTE$;DI:PRINT
460 PRINT"DISTANZA ORIZZ. PERCORSA
    = ";X1;" M":PRINT
470 PRINT"TEMPO RICHIESTO= ";T1;"
    SEC":PRINT
480 PRINT:PRINT:INPUT "VUOI VISUAL
    IZZARE TUTTI I DATI(S/N)";S$:I
    F S$="N" THEN 520
490 FOR I=1 TO I1
495 PRINT:PRINT:PRINT"DATI RELATIV
    I ALL'ISTANTE: ";T(I)
500 PRINT"X= "X(I),"Y= ";Y(I):PRIN
    T"Z= ";Z(I):PRINT
510 FOR K=1 TO 500:NEXTK:NEXTI
520 PRINT
541 REM ******************
543 REM * ROUTINE GRAFICA *
545 REM ******************
550 PRINT"[CLEAR]"
560 GOSUB 10020:I=0:GOSUB 1410
570 FOR I=1 TO I1:FOR J=1 TO 25:NE
    XTJ
580 GOSUB 1440:GOSUB 1410
590 NEXTI
592 REM *********************
593 REM * RITORNA ALL7INIZIO *
594 REM *    O TERMINA       *
595 REM *   L'ESECUZIONE     *
597 REM *********************
599 GET A$:IF A$="" THEN 599
600 POKE 53272,21:REM  CARATTERI N
    ORMALI
605 PRINT"[CLEAR]":INPUT "VUOI CON
    TINUARE(S/N)";S$:IF S$="N" THE
    N END
610 PRINT"[CLEAR]":GOTO 30
728 REM ****************
730 REM * SUBR.CALCOLO *
731 REM *  PARAMETRI   *
732 REM ****************
760 T(I+1)=T(I)+D
770 X(I+1)=X(I)+V*D
780 Z(I+1)=Z(I)-G*D
790 Y(I+1)=Y(I)+.5*(Z(I+1)+Z(I))*D
800 RETURN
1398 REM *****************
1399 REM *   SUBROUTINE   *
1400 REM *       DI       *
1401 REM * POSIZIONAMENTO *
1402 REM *****************
1410 PO=1024+(24-INT(Y(I+1)))*40+IN
     T(X(I+1)*CP)
1420 POKE PO,94:REM  POSIZIONA LA S
     FERA
1430 RETURN
1440 XP=X(I)*CP:YP=Y(I):GOSUB 20000
```

```
1445 PO=1024+(24-INT(YP))*40+INT(XP
     )
1450 POKE PO,CH:REM  POSIZIONE PREC
     EDENTE
1460 RETURN
10011 REM *************
10012 REM * CREAZIONE *
10020 REM *    DEI    *
10021 REM * CARATTERI *
10022 REM * SPECIALI  *
10023 REM *************
10024 REM * DISATTIVA *
10025 REM * KB E I/O  *
10027 REM *************
10031 POKE 56334,PEEK(56334) AND 254
     :POKE 1,PEEK(1) AND 251
10035 FOR I=0 TO 110:REM  CARATTERI
     DA COPIARE DALLA ROM
10036 FOR J=0 TO 7:REM  COPIA GLI 8
     BYTE DI UN CARATTERE
10037 POKE 12288+I*8+J,PEEK(53248+I*
     8+J):REM  COPIA UN BYTE
10038 NEXTJ:NEXTI:REM  PASSA AL PROS
     SIMO BYTE O CARATTERE
10039 POKE 1,PEEK(1) OR 4:POKE 56334
     ,PEEK(56334) OR 1:REM  ATTIVA
     I/O E KB
10040 POKE 53272,(PEEK(53272) AND 24
     0)+12:REM  IMPOSTA IL PUNTATOR
     E
10041 REM  CARATTERE ALLA LOCAZIONE
     DI MEMORIA 12288
10060 FOR CH=94 TO 110:REM  PROGRAMM
     A I CARATTERI DAL 94 AL 110
10080 FOR BY=0 TO 7:REM  COSTRUISCE
     GLI 8 BYTE DI UN CARATTERE
10100 READ  NU:REM  LEGGE 1/8 DEI DA
     TI DI UN CARATTERE
10120 POKE 12288+(8*CH)+BY,NU:REM  M
     EMORIZZA I DATI
10140 NEXTBY:NEXTCH:REM  PASSA AL PR
     OSSIMO BYTE O CARATTERE
10150 RESTORE :RETURN
10153 REM *****************
10154 REM * DATI PER CHRS *
10155 REM *   SPECIALI    *
10156 REM * DAL 94 AL 110 *
10158 REM *****************
10160 DATA  60,126,255,255,255,255,1
     26,60
10170 DATA  192,192,0,0,0,0,0,0
10180 DATA  48,48,0,0,0,0,0,0
10190 DATA  12,12,0,0,0,0,0,0
10200 DATA  3,3,0,0,0,0,0,0
10210 DATA  0,0,192,192,0,0,0,0
10220 DATA  0,0,48,48,0,0,0,0
10230 DATA  0,0,12,12,0,0,0,0
10240 DATA  0,0,3,3,0,0,0,0
10250 DATA  0,0,0,0,192,192,0,0
10260 DATA  0,0,0,0,48,48,0,0
10270 DATA  0,0,0,0,12,12,0,0
10280 DATA  0,0,0,0,3,3,0,0
10290 DATA  0,0,0,0,0,0,192,192
10300 DATA  0,0,0,0,0,0,48,48
10310 DATA  0,0,0,0,0,0,12,12
10320 DATA  0,0,0,0,0,0,3,3
20000 REM *************
20002 REM *   SCELTA   *
20005 REM * CARATTERE *
20006 REM *  GRAFICO  *
20007 REM *************
20010 IF (XP-INT(XP))<.25 THEN PX=1:
     GOTO 20050
20020 IF (XP-INT(XP))<.5 THEN PX=2:G
     OTO 20050
20030 IF (XP-INT(XP))<.75 THEN PX=3:
     GOTO 20050
20040 PX=4
20050 IF (YP-INT(YP))<.25 THEN PY=3:
     GOTO 20090
20060 IF (YP-INT(YP))<.5 THEN PY=2:G
     OTO 20090
20070 IF (YP-INT(YP))<.75 THEN PY=1:
     GOTO 20090
20080 PY=0
20090 CH=94+4*PY+PX:RETURN
21000 REM ****************
21030 REM * INTESTAZIONE *
21050 REM ****************
21070 PRINT"[3 DOWN]"
21074 PRINT"            %%%%%%%%%%%%%%%%
     %%%%%%%%"
21075 PRINT"            %[RVS]
                 [RVOFF]%"
21080 PRINT"            %[RVS] IL FENOME
     NO ELASTICO [RVOFF]%"
21085 PRINT"            %[RVS]
                 [RVOFF]%"
21086 PRINT"            %%%%%%%%%%%%%%%%
     %%%%%%%%"
21090 PRINT"[6 DOWN]                DI M
     AURO MASSETTI"
21100 FOR I=1 TO 5000:NEXTI:PRINT"[C
     LEAR]":RETURN
```

CROSS REFERENCE

di **Gloriano Rossi (i2KH)**

Una delle lacune del BASIC può essere identificata nella mancanza di un comando che permetta di sapere dove sia o dove venga citata una determinata variabile. Sui computer Commodore questa mancanza è in parte colmata, infatti è possibile caricare una routine in linguaggio macchina che normalmente viene identificata con il nome di FIND, e che però cerca una sola variabile per volta.

Ed ancora: il sistema di programmazione serio comanda che sia sempre presente una documentazione il più esplicita possibile relativa ad un qualsiasi programma o procedure. Poi, in fase di Up-dating, quella azione cioè che ci permette di correggere o aggiornare un programma, è estremamente utile avere una mappa delle variabili, in particolar modo se il programma in oggetto risulta particolarmente lungo.

Per queste ed altre valide ragioni, che non sto qui ad enunciare per non tediare ulteriormente il lettore, pubblico questo mio programma la cui prima versione data circa 5 anni fà. Nel 1979, abituato a lavorare sui grossi elaboratori (quelli che utilizzano linguaggi evoluti quali ad esempio il COBOL) ogni volta che si compilava un programma, (la compilazione è quella fase che permette di trasformare il linguaggio vicino all'uomo in una serie di codici vicini alla macchina), si otteneva una mappa delle variabili e delle labels (i numeri di riga richiamati in BASIC) con i relativi numeri di righe ove queste venivano citate.

Queste mappe si chiamavano proprio: Cross Reference.

Vediamo ora cosa accade nei grossi computers quando si dà loro in pasto un programma in linguaggio evoluto. Durante la fase di compilazione, il programma sorgente viene analizzato in ogni sua parte, dalla prima istruzione fino all'ultima. Questo fatto serve tra l'altro per controllare l'esattezza della stesura del programma, individuando gli eventuali errori di sintassi o di incongruenza.

Fatto ciò e... se tutto va bene, l'azione del compilatore prosegue con la trasformazione del programma sorgente in una nuova versione particolarmente vicina al linguaggio macchina. Sarà poi una nuova utility, chiamata LINKER, che provvederà a ritrasformare, o meglio aggiungere gli agganci, in maniera tale che dopo l'ultima manipolazione, il programma iniziale sia direttamente eseguibile dal computer. Come detto precedentemente, una delle opzioni del compilatore è quella di offrire al programmatore una mappa delle variabili e delle labels con le relative posizioni di dove vengono richiamate.

Questo mio articolo vuole rendere possibile anche per i possessori di Commodore avere una Cross Reference studiata appositamente per il BASIC. In un programma BASIC piuttosto breve l'uso del Cross Reference potrebbe risultare inutile, in quanto tutto è sott'occhio e con una sola occhiata si ha la situazione generale del programma in esame. Ma quando il programma oggetto incomincia avere una lunghezza davvero consistente...

Una variabile si trova qui, poi nell'altra pagina, poi ancora nell'ultima, ... ma non c'era anche da un'altra parte? Bhe, segna qui, poi la .... dopo un po' si ottiene un caos tale che non si capisce più niente. Con il Cross Reference tutto diventa più semplice: si va a vedere la variabile e quindi, dalla mappa stampata, si arriva direttamente in quelle posizioni univoche dove ad esempio A$ è citata.

ASCII		Codice attribuito		
da	a	variabili	linee	
1	33	4	4	
34		1	1	" doppi apici
35		4	4	x
36		7	4	$
37		7	4	%
38		7	4	&
39		4	4	,
40		8	4	(
41	47	4	4	
48	57	6	6	numeri
58	64	4	4	: ; (—) ?
65	90	5	4	lettere
91	130	4	4	[] — (END/FOR/NEXT)
131		3	3	(DATA)
132	142	4	4	(comandi BASIC)
143		2	2	(REM)
144	255	4	4	(comandi BASIC)

Figura 1. Tabellina di attribuzione dei codici ai relativi valori ASCII dei caratteri presenti su ogni riga BASIC.

```
-------*------------------ - - -
VAR.   ! LINEA DEL PROGRAMMA
-------*------------------ - - -
       !
A      !  310   312   328   342   344   346
       !  380   382
A$     !  226   292   294   306   308   310
       !  312   342   344   394   402   404
       !  406   408   410   414   462   466
       !  468   470   472
A$(    !  162   264   266   358   360   364
       !  370
B      !  256   262   276   280   354   356
       !  360   362   368
B$     !  226   246   292   294   398   406
       !  410   412   470   474   480
B$(    !  162   170   172   210   398
C      !  314   372   380   384   386   388
       !  390   392   430   434   436
C$     !  400   412   414   472   494
C(     !  162   188   190   194   196   198
       !  200   346
C1     !  316   348   374   386   390
C2     !  184   186   194   314   350   372
       !  378   392   464
C9     !  346   348   384   386   390   396
       !  398
J      !  188   190   194   262   264   266
       !  270   272   274   356   358   360
       !  362   460   462   498
K      !  260   270   272   352   360   362
       !  364   366   370   464   466   468
       !  470   472
L      !  306   308   328   330
L$     !  258   268   330   332
M$     !  350   358   360   370   376   392
       !  394
MA$    !  158   450   454
MB$    !  160   452
MC$    !  458   476   496
P$     !  204   206   208   210   446
Q$     !  166   170   408
S$     !  168   332   482   492   494
X      !  260   276   460
X$     !  266   268   270   272   456   474
       !  480   482
X$(    !  162   270   272   462
Y      !  478   484   486   488
Z      !  432   434   486
Z$     !  176   178   186   428   434
```

Figura 2. Il programma CROSS REFERENCE esaminando se stesso ed inizializzato secondo le variabili produce un foglio esattamente in questa maniera. Osservate che i nomi delle variabili e degli array sono messi in ordine alfabetico e gli array sono distinguibili dalla variabile per la presenza della parentesi aperta.

Il programma

Il programma Cross Reference è stato in un primo momento realizzato in una forma un po' 'scialba', povera. È stato in seguito l'interesse e l'utilità che mi hanno spinto a migliorarlo fino ad ottenere l'edizione che vi propongo. Non esistono grosse limitazioni nell'utilizzo di Cross Reference, in quanto può girare tranquillamente su tutti i Commodore che abbiano sufficiente spazio in memoria, ecco perché nella presentazione del listato l'ho dichiarato inaccessibile per il VIC 20 versione base, infatti questo home potrà sfruttare il CR solo con almeno una espansione di memoria. Per ciò che concerne poi i nuovi Commodore (C16 e Plus-4) si hanno delle istruzioni BASIC in più e quindi non ho ritenuto di ampliare la logica di esame (elaborativa), almeno per questo momento.

Il listato

Il listato del programma prevede numerosi commenti che illustrano passo passo l'evoluzione delle istruzioni. Una particolare attenzione si può focalizzare là dove è necessaria la personalizzazione secondo il tipo di Commodore utilizzato.

Vediamo la riga **246**. Questa riga dovrà essere battuta se si utilizza un VIC20, un C16 oppure il Plus-4. Se si utilizza una serie Commodore 3000, 4000 o 8000, è necessario cambiare il numero 8 con il numero 4; infatti la riga in esame controlla se il programma oggetto inizi alla locazione di partenza del BASIC. Sappiamo infatti che nel Commodore 64 detta locazione corrisponde a $0801, mentre nella serie 3000 - 4000 - 8000 la locazione è $0401. Una seconda personalizzazione potrà essere eseguita nella zona di programma dedicata alla stampa.

La routine proposta prevede la possibilità di eseguire la visualizzazione dei risultati sia su stampante che su video. Per una giusta impaginazione su uno schermo a 40 colonne, la variabile Z (riga **432**) dovrà corrispondere al numero 6, mentre se l'output dovesse andare su stampante questo numero potrà essere modificato fino ad un massimo di 16 (riga **434**).

Come si usa

Vediamo come poter utilizzare il nostro Cross Reference una prima volta in-

```
PROGRAMMA : CROSS REFERENCE

------*--------------- - - -
VAR.  ! LINEA DEL PROGRAMMA
------*--------------- - - -
      !
  174 !   182
  184 !   178
  202 !   206
  256 !   396    404
  270 !   270
  292 !   256
  342 !   396    406
  350 !   350
  368 !   354    360
  372 !   358
  378 !   348
  386 !   378
  394 !   390
  396 !   380    382    384    388    392
  398 !   396
  402 !   408    416
  418 !   294
  470 !   464
  484 !   474
  494 !   486
```

Figura 3. Eseguendo ancora una volta il CROSS REFERENCE su se stesso avendo però inizializzato secondo le linee, troveremo un risultato fatto in questa maniera.

sieme. Abbiamo terminato di digitare il CR e subito eseguiamo il salvataggio sul dischetto (SAVE "CROSS REFERENCE", 8). Fatto ciò eseguiamo il RUN.

La prima domanda a cui dobbiamo rispondere è quella relativa al tipo di esame: per variabili o per linee. Proviamo una prima volta con le variabili. Tenete presente che rispondendo in questa maniera vengono analizzate non solo le variabili, ma anche gli array.

La seconda domanda che CR ci chiederà sarà quella relativa al programma da prendere in esame e noi, rispondendo con CROSS REFERENCE non faremo altro che dirgli di esaminare se stesso. A questo punto CR farà tutto da solo: accederà al discetto ed incomincerà a leggere il programma richiesto byte per byte, linea per linea, fino alla fine.

Mentre CR esegue questo lavoro di esame potrete vedere il proseguire della elaborazione sul video, infatti ogni riga che contenga una variabile (o numero di linea) viene visualizzata sullo schermo con i relativi nomi di variabili (o numeri di linee).

Dopo aver terminato l'esame del programma, ecco l'ultima domanda: su stampante?

È chiaro che rispondendo affermativamente non sarà visualizzato alcun dato sul video, ma tutto verrà stampato su carta. Questo fatto avviene perché la routine di output sostanzialmente è la medesima, mentre cambia esclusivamente il numero di device. Infatti 3 corrisponde al video e 4 (o 5 per alcune stampanti) equivale alla stampante.

```
100 REM ************************
102 REM *                      *
104 REM *      C R O S S       *
106 REM *                      *
108 REM *   R E F E R E N C E  *
110 REM *                      *
112 REM ************************
114 REM * AUTORE :    I 2 KH   *
116 REM *     GLORIANO ROSSI   *
118 REM ************************
120 REM * VIC20             NO *
122 REM * VIC20 EXP         SI *
124 REM * COMMODORE 64      SI *
126 REM * C16              *NO *
128 REM * PLUS 4           *NO *
130 REM * COMMODORE 3000    SI *
132 REM * COMMODORE 4000    SI *
134 REM * COMMODORE 8000    SI *
136 REM * COMMODORE 3000    SI *
138 REM * COMMODORE 4000    SI *
140 REM ************************
142 REM * IL C16 ED IL PLUS 4  *
144 REM * POSSONO UTILIZZARE   *
146 REM * QUESTO PROGRAMMA SOLO*
148 REM * SE SI CAMBIANO ALCUNI*
150 REM * VALORI DELLA TABELLA *
152 REM * C< .                 *
154 REM ************************

156 REM
158 MA$="------*---------------- -
   - -"
```

```
160 MB$="VAR.   ! LINEA DEL PROGRAM
    MA "
162 DIM  A$(15),B$(3),X$(500),C(25
    5)
164 PRINT"[CLEAR]C R O S S - R E F
     E R E N C E[2 DOWN]"
166 Q$=CHR$(34):REM  VIRGOLETTE
168 S$="         "
170 B$(1)=Q$:REM  VIRGOLETTE
172 B$(3)=CHR$(58):REM  DUE PUNTI
174 PRINT"[DOWN]LISTA [RVS]V[RVOFF
    ]ARIABILI A [RVS]L[RVOFF]INEE
      V[3 LEFT]";
176 INPUT Z$
178 IF Z$="V" OR Z$="L" THEN 184
180 PRINT"[3 UP]"
182 GOTO 174
184 C2=5
186 IF Z$="L" THEN C2=6
188 FOR J=1 TO 255:C(J)=4:NEXTJ:RE
    M  INIZIALIZZAZIONE DI VALORE
    COSTANTE
190 FOR J=48 TO 57:C(J)=6:NEXTJ:RE
    M  PER I CODICI RELITIVI AI NU
    MERI
192 REM  SE ELAB PER VARIABILI 5=L
    ETTERE, 7=$%&, 8=(
194 IF C2=5 THEN FOR J=65 TO 90:C(
    J)=5:NEXTJ:FOR J=36 TO 38:C(J)
    =7:NEXTJ:C(40)=8
196 C(34)=1:REM  CODICE PER VIRGOL
    ETTE
198 C(143)=2:REM  COD PER REM
200 C(131)=3:REM  COD PER DATA
202 PRINT"[DOWN]NOME PROGRAMMA      *
    [3 LEFT]";
204 INPUT P$
206 IF P$="*" THEN PRINT"[3 UP]":G
    OTO 202
208 OPEN 1,8,3,"0:"+P$+",P,R"
210 PRINT"[DOWN][RVS]OPEN1,8,3,"B$
    (1)"0:"P$",P,R"B$(1)"[RVOFF][D
    OWN]":REM  VISUALIZZA CIO' CHE
     FA
212 REM ********************
214 REM * PRENDE I DUE     *
216 REM * PRIMI BYTE CHE   *
218 REM * IDENTIFICANO     *
220 REM * LA LOCAZIONE DI  *
222 REM * INIZIO BASIC     *
224 REM ********************
226 GET #1,A$,B$
228 REM ********************
230 REM * VIC20      NO  246 *
232 REM * C64            <>8 *
234 REM * 3000 !             *
236 REM * 4000 !-->      <>4 *
238 REM * 8000 !             *
240 REM * C16        NO  246 *
242 REM * PLUS 4     NO  246 *
244 REM ********************
246 IF ASC(B$)<>8 THEN CLOSE 1:STO
    P
248 REM ********************
250 REM * INIZIO SCANSIONE *
252 REM * PROGRAMMA OBJ    *
254 REM ********************
256 IF B=0GOTO 292
258 PRINTL$;
260 K=X
262 FOR J=B TO 1 STEP -1
264 :PRINT" ";A$(J);
266 :X$=A$(J)
268 :X$=X$+L$
270 :IF X$(K)>X$ THEN X$(K+J)=X$(K
    ):K=K-1:GOTO 270
272 :X$(K+J)=X$
274 NEXTJ
276 X=X+B
278 PRINT
280 B=0
282 REM ********************
284 REM * ESAME DELLA      *
286 REM * NUOVA RIGA BASIC *
288 REM * TEST DI FINE     *
290 REM ********************
292 GET #1,A$,B$:REM  PUNTATORE A
    PROSSIMA RIGA BASIC
294 IF LEN(A$)+LEN(B$)=0GOTO 418:R
    EM  EQUIVALE A FINE PROGRAMMA
296 REM ********************
298 REM * PRENDE IL NUMERO *
300 REM * DELLA RIGA BASIC *
302 REM * IN ESAME         *
304 REM ********************
306 GET #1,A$:L=LEN(A$)
308 IF L=1 THEN L=ASC(A$)
310 GET #1,A$:A=LEN(A$)
312 IF A=1 THEN A=ASC(A$)
314 C=C2
316 C1=-1
318 REM ********************
320 REM * TRASFORMAZIONE   *
322 REM * NUMERO DI RIGA   *
```

```
324 REM * IN STRINGA          *
326 REM ********************
328 L=A*256+L
330 L$=STR$(L)
332 IF LEN(L$)<6 THEN L$=LEFT$(S$,
    6-LEN(L$))+L$
334 REM ********************
336 REM * ESAME DELLA ZONA *
338 REM * BASIC DELLA RIGA *
340 REM ********************
342 GET #1,A$:A=LEN(A$)
344 IF A=1 THEN A=ASC(A$)
346 C9=C(A)
348 IF C9>C1GOTO 378
350 IF C2=6 AND LEN(M$)<5 THEN M$=
    " "+M$:GOTO 350
352 K=0
354 IF B=0GOTO 368
356 FOR J=1 TO B
358 IF A$(J)=M$GOTO 372
360 IF A$(J)<M$ THEN NEXTJ:K=B:GOT
    O 368
362 FOR K=B TO J STEP -1
364 A$(K+1)=A$(K)
366 NEXTK
368 B=B+1
370 A$(K+1)=M$
372 C=C2
374 C1=-1
376 M$=""
378 IF C2=5GOTO 386
380 IF A=137 OR A=138 OR A=141 OR
    A=167 THEN C=6:GOTO 396
382 IF A=44 OR A=32GOTO 396
384 IF C9<>6 THEN C=9:GOTO 396
386 IF C9=C THEN C=-1:C1=4
388 IF C>6GOTO 396
390 IF C<0 AND C9>C1 AND C9>6 THEN
     C1=C9:GOTO 394
392 IF C2=5 THEN IF LEN(M$)>2 OR C
    >0GOTO 396
394 M$=M$+A$
396 ON C9+1GOTO 256,398,398,398:GO
    TO 342
398 B$=B$(C9)
400 C$=""
402 GET #1,A$
404 IF A$=""GOTO 256
406 IF A$=B$GOTO 342
408 IF A$<>Q$GOTO 402
410 A$=B$
412 B$=C$
414 C$=A$
416 GOTO 402
418 CLOSE 1
420 REM ********************
422 REM * ROUTINE DI          *
424 REM *    S T A M P A    *
426 REM ********************
428 INPUT "SU STAMPANTE     S[3 LEF
    T]";Z$
430 C=3:REM  DEVICE VIDEO
432 Z=6:REM  GRUPPI PER LINEA-1
434 IF Z$="S" THEN C=4:Z=7:REM  SU
     STAMPANTE, GRUPPI DA 6 A 15 (
    A PIACERE)
436 OPEN 2,C
438 PRINT#2
440 PRINT#2,"[CLEAR]"
442 PRINT#2,"C R O S S   R E F E R
     E N C E"
444 PRINT#2
446 PRINT#2,"PROGRAMMA : ";P$
448 PRINT#2
450 PRINT#2,MA$
452 PRINT#2,MB$
454 PRINT#2,MA$
456 X$=""
458 MC$="      !"
460 FOR J=1 TO X
462 A$=X$(J)
464 IF C2=6 THEN K=6:GOTO 470
466 FOR K=1 TO LEN(A$)
468 IF MID$(A$,K,1)<>" " THEN NEXT
    K:STOP
470 B$=LEFT$(A$,K-1)
472 C$=MID$(A$,K+1)
474 IF X$=B$GOTO 484
476 PRINT#2,MC$
478 Y=0
480 X$=B$
482 PRINT#2,X$;LEFT$(S$,5-LEN(X$))
    " !";
484 Y=Y+1
486 IF Y<ZGOTO 494
488 Y=1
490 PRINT#2
492 PRINT#2,S$" !";
494 PRINT#2,LEFT$(S$,6-LEN(C$)),C$
    ;
496 MC$=""
498 NEXTJ
500 PRINT#2
502 CLOSE 2
```

Se vuoi abbonarti

Registrate il mio abbonamento annuale a Commodore.

☐ Ho versato oggi stesso il canone di Lire 25.000 a mezzo c/c postale n° 31532203 intestato a: Commodore Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

☐ Accludo assegno per lire 25.000 banca ______ n° ______ a favore di Commodore Systems Editoriale

Il mio computer è: VIC 20 ☐, C 64 ☐, altro (specificare) ______

Ho ☐ / non ho ☐ la stampante, ma voglio ☐ comprarla.

Preferisco programmi di gioco ☐, didattici ☐, d'utilità ☐, altro ______

Nome ______ Cognome ______

Via ______ n° ______ CAP. ☐☐☐☐☐ Città ______

Tel. ______

Se vuoi collaborare

Registratemi fra i collaboratori regolari di Commodore.

A titolo di prova vi invio un articolo e la cassetta col programma "______ ______" di cui vi garantisco l'assoluta originalità autorizzandovene la pubblicazione.

☐ Scrivetemi all'indirizzo sottoindicato ______

Nome ______

Via ______ N° ______

Tel. ______ CAP ______ Città ______

Se vuoi un consiglio o consigliarci

HELP ______

Nome ______

Via ______ n° ______ CAP. ☐☐☐☐☐ Città ______

Tel. ______ Orario ______

Il mio computer è configurato:

			Sono in possesso	No	Ho intenzione di acquistare
Vic 20 ☐	espanso a ___ K		☐	☐	☐
C 64 ☐			☐	☐	☐
Floppy ☐	quale: 1541 ☐	altro ______	☐	☐	☐
Stampante ☐	quale: MPS801 ☐	altro ______	☐	☐	☐
Plotter ☐	quale: 1520 ☐	altro ______	☐	☐	☐
Registratore ☐	quale: 1530 ☐	altro ______	☐	☐	☐

Televisore ☐, TV-Monitor ☐, Monitor ☐, Colore ☐, B/N ☐

Nome ______ Cognome ______

Via ______ n° ______ CAP. ☐☐☐☐☐ Città ______

Tel. ______

Se vuoi vendere o comprare

Vendo ☐ Compro ☐ ______

Nome ______

Via ______ n° ______ CAP. ☐☐☐☐☐ Città ______

Tel. ______ Orario ______

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si, voglio abbonarmi

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si, voglio collaborare

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si, chiedo consiglio

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si, voglio votare

Da inviare in busta chiusa a:

Spett.le rivista
Commodore
Systems Editoriale

v.le Famagosta, 75
20142 Milano

Si vendo/ compro

Scambiatevi le liste

Tutte le richieste di Vendo-Scambio-Compro programmi per... verranno citate esclusivamente con il solo nome, cognome ed indirizzo. Sta a voi lettori scambiarvi le varie liste. Verranno pubblicati integralmente a giudizio del caporedattore, quegli annunci che non fanno parte della categoria di cui sopra. Buoni scambi e vendite!!

(Gloriano Rossi)

Enrico Mascioletti, Via Rimembranze, 11 - 43100 Parma - Tel. 0521/582656

Antonio Agus - Via Sant'Antonio, 31 - 09016 Iglesias - Tel. 0781/22423

Stefano Ceci - Via Paladino, 8 - 80100 Napoli - 081/205602

Livio Modena - Via Alberghi Sciacca Mare - 92019 Sciacca - Tel. 0925/93000

Dario Fumagalli - Via Boccaccio, 32 - 20145 Milano

Stefano Dietrich - Largo Tevere P. Papa, 183 - 00146 Roma - Tel. 06/5562629

Silvano Bompieri - Via Baccaglioni, 8 - 46040 Monzabano (MN) - Tel. 0376/845372

Matteo Doveri - Via I Maggio, 15 - 56025 Pontedera (Pisa) - Tel. 0587/212154

Gianni Gaudino - Via Graglia, 18 - 10136 Torino - Tel. 011/352830

Leonardo Landini - Via Corcos, 5 - 50100 Firenze - Tel. 055/14360

Roberto Cavataio - Via G. Agnelli, 1 - 10070 Robassomero (Torino) - Tel. 011/9235700

Riccardo Lorusso - Viale Montello, 5 - 20154 Milano - Tel. 02/6556024

Flavio Mambelli - Via G. Valmarana, 68 - 00139 Roma - Tel. 06/8122497

Mauro Iannucci - Via E. Fermi, 49 - 00146 Roma

Mauro Miccinilli - Via Giuseppe Armzllini, 21 - 00143 Roma - Tel. 06/5917903

Francesco Tosi - Via Di Trasone, 6 - 00199 Roma - Tel. 06/8381140

Assunta Lama - Via G. Rossini, 4 - 19100 La Spezia

Paolo Vergoni - Via Appia, 69 - 06100 Perugia - Tel. 075/66918

Tonino Crasto - Via Don Bosco, 9/e - 80141 Napoli - Tel. 081/449915 - 081/7512140

Roberto Mancuso - Via F. Magellano, 19 - 80038 Pomigliano d'Arco (Na) - Tel 081/8841291

José Andreani - Via Antinori, 8 - 60100 Ancona - Tel. 895165

Luca Perini - Lungo Castellano, 44 - 63100 Ascoli Piceno - Tel. 0736/52175

Paolo Caron - Borgo Padova, 81 - 35013 Cittadella - Tel. 049/591785

Massimo Leoncini - Casella Postale 85 - 16135 Rapallo (Genova)

Mariano Talamo - Via Colucci, 10 - 70019 Triggiano - Tel. 080/681470

Giorgio Cavicchioli - Via F. Turati, 14 - 41012 Carpi

Franco Fantoni - P.O. BOX 259 - 51100 Pistoia

Alessandro Bruno - Via Alessandria, 6 - 15033 Casale Monferrato - Tel. 0142/79784

Marcello Cegaro - Via Vittorio Emanuele, 70 - 81030 Parete (Ce) - Tel. 071/8117213

Daniele Jommi - Via Macciotta, 8 - 07041 Alghero (SS)

Maurizio Maramo - Via Col Berretta, 15 - 35100 Padova - Tel. 049/656408

Enrico Carrara - Via Tomavo, 12 - 00195 Roma

Dino Tigli - Via Pergola, 67 - 22053 Lecco - Tel.)341/366172

Giulio Ravagni - Casella Postale 224 - 38068 Rovereto (TN) - Tel. 0464/34475

Luca Tonetto - Padova - Tel. 049/27167

Commodore Club - Via Filisto, 37 - 96100 Siracusa - Tel. 0931/35866

Carmen Carucci - Via Calamandrei, 1 - 14049 Nizza Monferrato - Tel. 0141/727216

Franco Ferri - Via S. Lazzaro, 31 - 40068 San Lazzaro (BO) - Tel. 051/461504

Walter Ferrero - Via Assietta, 5/3 - 10098 Rivoli (TO) - Tel. 011/9587985

Enrico Cipriani - Via del Dominichino, 19 - 61032 Fano (PS) - Tel. 0721/874688

Alin Nadin - Shany Club - Via Passeggiata, 39 - 30020 Bibione (VE) - Tel. 0431/43233

Maurizio Marano - Via Gozzadini, 6 - 40055 Castenaso (BO) - tel. 051/781679

Massimo Capelli - Via Leon Alberti, 64 - 40139 Bologna - Tel. 051/306382

Marcello Clarizia - V.le Garibaldi, 8 - 84013 Cava dei Tirreni

Maurizio Cottini - Via Alfieri, 5 - 27029 Vigevano (PV)

Vincenzo Musicò - Via Paolo Blondino, 12 - 98100 Messina - Tel. 090/2938626

Mario Fanelli - Via Trento, 1 - 20060 Cassina de' Pecchi (MI) - Tel. 02/9528793

Sebastiano Caramagno - Via Cont. Cipollazzo - 96011 Augusta (SR) - Tel. 0931/993333

Luca Lauro - Via Vittorio Colonna, 220 - 8077 Ischia (NA) - Tel. 081/992589

Nicola Di Piazza - Via Dol Dominichino, 23 - 61032 Fano (PS) - 0721/877327

Sergio Rivoletti - Via Tevere - 63017 Porto San Giorgio (AP) - Tel. 0734/4419

Gianni Mazzesi - Via Cella, 329 - 48020 S. Stefano (RA) - Tel. 0544/573529

Bruno Bursi - Via Delle Vigne, 87 - 00148 Roma - Tel. 06/5238575

Paolo Pavoni - Via C. Battisti, 111 - 64011 Alba Adriatica (TE) - Tel. 0861/72191

Emanuele Bertodo - Via Ponchielli, 7 - 20129 Milano - Tel. 02/2719611

Enzo Petrizzo - Via Nazionale, 157 - 84030 Padula Scalo - Tel. 0975/74022

Domenico Dieci - Via Castelfidardo, 2/2 - 42100 Reggio Emilia - Tel. 0522/43928

Maurizio Maestri - Via P.G. Genocchi, 492 - 47023 Cesena (FO)

Andrea Zampedroni - Via Beolchi, 4 - 20151 Milano

Carlo Abrunzo - Via P. Collenuccio 5/A - 80136 Napoli - Tel. 081/214667

Giovanni Giamminola - Via Vittorio Veneto, 14 - 22070 Appiano G. (Como) - Tel. 933176

Cesare Giarda - Via Viscerei, 33 - 28065 Cerano (NO) - Tel. 0321/728062

Marzio Maretti - Via G. Rossini, 21 - 50144 Firenze - Tel. 055/364766

Giovanni Abbate - Via Medaglie d'oro, 87 - 97015 Modica (RG) - Tel. 0932/944256/900292

Calcedonio Meli - Via G.A. Valenti, 89 - 93100 Caltanissetta - Tel. 0934/23365

Massimo Selleri - Via Don Minzoni, 72 - 45100 Rovigo

Alfredo Casciano - Via Mons. Virgilio, 105 - 85029 Venosa (PZ) - Tel. 0972/32915

Matteo Gianera - Via Lago di Garda, 41 - 48100 Ravenna - Tel. 0544/34988

Paolo Sommavilla - Via Forcellini, 3 - 35127 Padova - Tel. 049/757046

Alessandro Paolini - Via Marcello Provenzale, 9 - 00168 Roma - 06/6278922

Roberto Sant - Via Diaz, 18 - 33010 Trieppo Grande (Udine) - Tel. 0432/960604

Antonio Mandalà - Via Interna, 40 - 33170 Pordenone - Tel. 0434/32231

Luca Bernardini - Via S. Martino, 21 - 05100 Terni - Tel. 0744/56870

Riccardo Maffioli - Via Abruzzi, 97 - 25100 Brescia

Maurizio Caruso - Viale Libertà, 85 - 95014 Giarre (CT) - Tel. 095/932723

Luca Naccarato - Via Trieste, 21 - 87068 Rossano Scalo (CS) - Tel. 0983/21451

Massimo Tolone - Via G. Galilei, 16 - 35043 Monselice (PD) - Tel. 0429/74033

Paolo Gallo - Via Garibaldi, 83 - 96014 Floridia (SR) - Tel. 0931/941056

Franco Radice - Via G. Ingegnoli, 25 - 20131 Milano - Tel. 02/2899553

Alberto Borgini - Via San Rocco, 34 - 21031 Gallarate (MI) - Tel. 0331/795770

Fabrizio Parenti - Via Prampolini, 5 - 43100 Parma - Tel. 0521/72924

Guido Gafforelli - Viale Settala, 1 - 20124 Milano

Valerio Cristiani - Via Alcide de Gasperi, 49 - 63016 P.to P. Picena - Tel. 0733/688117

Renato Menegatti - Via Piemonte, 26 - 20052 Monza (MI)

Sergio Innocente - Via Saccardo, 16 - 31100 Treviso - Tel. 0422/51182

Giancarlo Giuliano - Via Carso, 3 - 18039 Ventimiglia - Tel. 0184/34463

Stefano Metalli - Via Campagnano, 56 - 00189 Roma - Tel. 06/3271446

Roberto Cotza - Via Puccini, 60 - 20099 Sesto San Giovanni - Tel. 02/2425392

Gianluca Amoroso - Via G. Di Vittorio, 7/13 - 30170 Mestre (VE) - Tel. 041/612015

Ivan Visentini - Via I. Nievo, 11 - 33100 Udine - Tel 0432/21363

Tommaso Mazzoni - Via Ludovico Cardi, 9 - 50053 Empoli (FI)

Maria Antonia Monti - Casella Postale 45 - 55052 F. Barga (Lucca)

Michele Petracca - Via Donatello, 12 - 35027 Noventa P. (PD) - Tel. 049/627164

Romolo Moragli - Via Benevento, 20 - 07026 Olbia (SS) - Tel. 0789/21803

Ezio Pitton - Via Zorutti, 10 - 33070 Casarsa (PN) - Tel. 0434/869950.

Fabio Bozzato - Via S. Anna, 213/H - Santa Anna di Chioggia (VE) - Tel. 041/4950397

Andrea Fabrello - Via Div. Julia, 20 - 36011 Arsiero (VI) - Tel. 0445/71348

Federico Franchini - Via Covignano, 218 - 47037 Rimini - Tel. 0541/770792

Rosalba Balducci - Via Baccarani, 7 - 60100 Ancona - Tel. 071/55089

Donato Domus - Via Crimea, 3 - 20147 Milano - Tel. 02/4049179

Vincenzo Cioffi - II Traversa Nicolardi, 32 - 80131 Napoli - Tel. 081/7434424

Marco Saltarelli - Via XXIV Maggio, 7 - 36054 Montebello - tel. 0444/749024

Vincenzo Todaro - Via Carrubbella, 155 - 95030 Gravina (CT) - Tel. 095/414429

Fabrizio Prudente - Via L. Tripoti, 7/a - 64100 Teramo - Tel. 0861/411184

Claudio Giovanelli - Via Ripamonti, 194 - 20141 Milano - Tel. 02/536926 uff. 02/563105

Vendo Floppy 1541 a L. 550.000 *(Fabio Galeazzi - Via P.B. Farinelli, 10 - 60014 Falconara - AN - Tel. 071/9173656 dopo le 20).*

Vendo Commodore 64 chi ha il manuale d'uso in inglese e vuole la traduzione in italiano cambio con giochi o utlity (o vendo a L. 10.000 solo per chi non ha cambio) *(Giogio Gualtieri - Via Bolzano, 29/A - 20127 Milano - Tel. 02/2842565).*

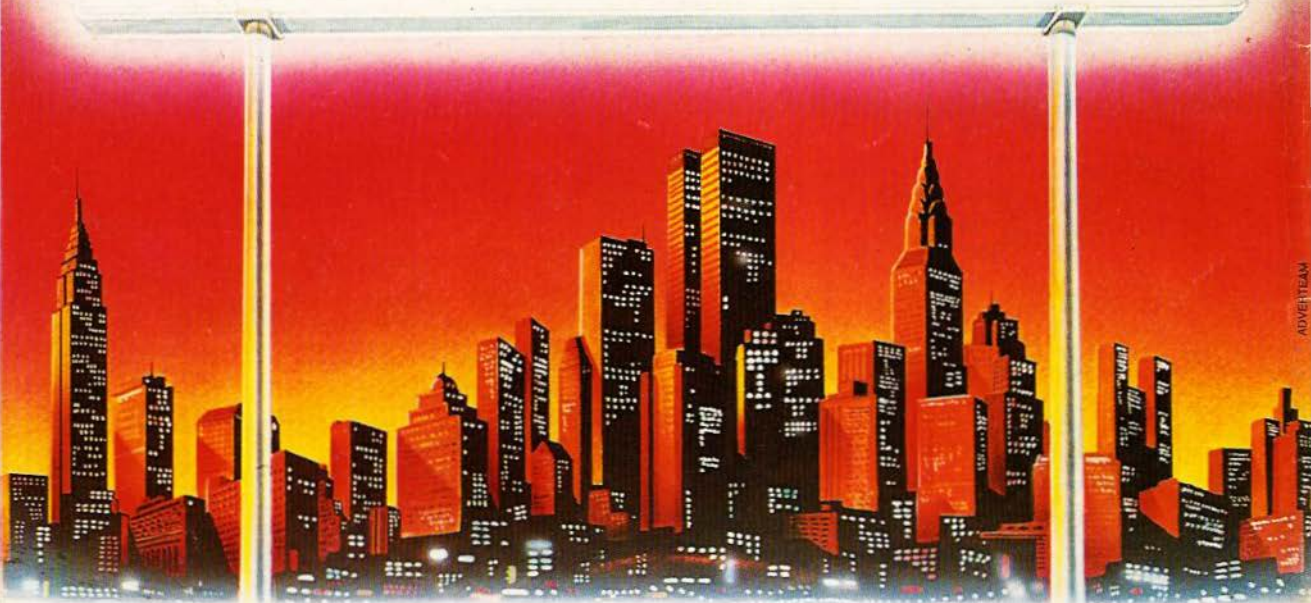
Vendo Commodore Vic 20, tre mesi di vita + joystick + 8 cartucce giochi + introduzione Basic 1° e 2° parte con relative 4 cassette + una trentina di giochi su cassetta. Il tutto a L. 300.000 *(Ferdinando Fontand - Via Pavia, 51 - 15100 Alessandria - Tel. 0131/343086 - 66604 dopo le ore 20).*

Vendo regalo Vic 20 + 2 cartucce giochi + scacchi a chi acquista in blocco Vic 1111 - 16K Ram + Vic 1211 - 3K super expander + Vic 1212 - programmers Aids + Vic 1020 cabinet espansione 6 Slot. Valore totale oltre L. 700.000 vendo a L. 400.000 *(Dario Giambelli - Via De Cristoforis, 15 - 20124 Milano - Tel. 02/6572867 ore 19/22).*

Vendo 1° e 2° volume dell'enciclopedia BASIC al modico prezzo di L. 40.000 *(Stefano De Blasis - Via Corticella, 16 - 40128 Bologna - Tel. 051/362839 ore 14-15 - 20/22).*

Vendo Videogioco Philips G7400 con 5 relative cartucce (sci, baseball, introduzione alla programmazione, ecc.). Con una ulteriore piccola spesa lo si può far diventare computer con 22K Ram. Il tutto a L. 280.000. Trattabili. Pagamento in contrassegno. *(Andrea Navaglia - Salita al Poggio, 52 - 18030 Poggio di Sanremo - Tel. 0184/889393 ore pasti).*

Vendo il libro "Voi e il vostro Commodore 64" a L. 15.000. Compro "Guida di riferimento per il programmatore CBM 64". Telefonare o scrivere. *(Luca Menegozzo - Via B. Scardeone, 6 - 35012 Camposanpietro - PD - Tel. 049/5790983 al pomeriggio).*

Computer School

L'esperienza insegna.

La richiesta di corsi d'informatica cresce ogni giorno di più. Ma non basta conoscere bene il computer per saperne insegnare l'uso ad un pubblico tanto ansioso d'apprendere quanto privo di qualsiasi nozione di base in materia. Nè la sola esperienza didattica è sufficiente per entrare in questo settore. Così, incalzato dalla domanda, anche tu che non vuoi rispondere con un insegnamento insufficiente o improvvisato, probabilmente sei alla ricerca d'una metologia provata e sicura, di una manualistica coerente e completa, di sussidi audiovisivi e schemi di lavoro. Computer School [R] possiede una solida, tangibile e collaudata esperienza d'insegnamento dell'informatica e ti dà tutto questo, insieme al know-how ed al supporto necessario perchè anche tu possa entrare con successo in questo promettente mercato. Inoltre, consentendoti di presentarti agli utenti potenziali con il suo marchio e la sua insegna, ti offre un ulteriore vantaggio: beneficiare d'una possente azione pubblicitaria sulle più diffuse e prestigiose pubblicazioni del settore. Se perciò vuoi essere la prima Computer School della tua città non esitare a contattarci.

Franchising per insegnare.

20090 Trezzano S/N (MI) - V.le C. Colombo, 49 - Tel. (02) 4454352/4459252